CA-IDMS®/DB Audit

User Guide
15.0

Computer Associates™

First Edition, December 2000

Contents

About This Guide

Purpose

This guide provides the necessary information to run CA-IDMS/DB Audit. In addition, the many features that CA-IDMS/DB Audit offers are documented to assist you.

Organization

Chapter	Description
1	Presents a summary of CA-IDMS/DB Audit features and functions.
2	Provides an overview of the types of errors CA-IDMS/DB Audit can find. It compares the Standard and QuickCheck auditing methods, and presents information on the processes used in the Audit and Fix phase.
3	Presents CA-IDMS/DB Audit parameter statements and options. Conventions and syntax are summarized in figure format and a detailed discussion is given for each parameter option.
4	Describes and gives examples of the two types of reports that are available with CA-IDMS/DB Audit.
5	Describes auditing methods and operational procedures. It provides information for determining storage requirements, and information for operating CA-IDMS/DB Audit efficiently.
6	Provides a list of all informative, error, and warning messages generated by CA-IDMS/DB Audit, along with reasons for occurrence and suggested actions to be taken.
Index	Provides an alphabetical list of CA-IDMS/DB Audit concepts with their locations in the user guide.

CA-IDMS/DB Audit Publications

In addition to this guide, Computer Associates supplies the following documentation:

Name	Contents
CA-IDMS installation guides	An installation guide is provided to use as a reference tool and gives complete information about the installation of the products.

Chapter 1. General Information

CA-IDMS/DB Audit is a software tool that allows you to examine the physical integrity of all or part of a CA-IDMS database. In addition, you can direct CA-IDMS/DB Audit to fix most of the errors found and to report the errors and the corrections. A report of simulated corrections can be examined before actually updating the database. Elimination of physical inconsistencies in your database sets the stage for smooth operations of application programs.

1.1 Physical Integrity of the Database

The physical integrity of a database is a primary concern of the database administrator. If there are physical inconsistencies in the data, the system programmers, internal auditors, applications programmers, and data processing managers cannot use programs effectively, because application programs will not accurately retrieve the records from the database.

1.2 CA-IDMS/DB Audit Finds Physical Integrity Errors

CA-IDMS/DB Audit finds physical integrity errors in your CA-IDMS database. Several errors are illustrated in Figure 1.1. CA-IDMS/DB Audit detects these problems:

- Inconsistent page headers and footers
- Record occurrences that fall outside the page range for a record type
- Set disconnections
- Invalid pointers
- Fragmented records that cannot be properly assembled
- Orphan records not properly connected to a set
- Integrated index structure errors
- Set integrity errors
- Logically deleted records (LDELs)

1.3 CA-IDMS/DB Audit Fixes Physical Integrity Errors

CA-IDMS/DB Audit can fix most of the errors it finds; it can also physically delete LDELs and adopt integrated index orphans. If you want to review the corrections before actually updating the database, CA-IDMS/DB Audit produces reports of simulated changes.

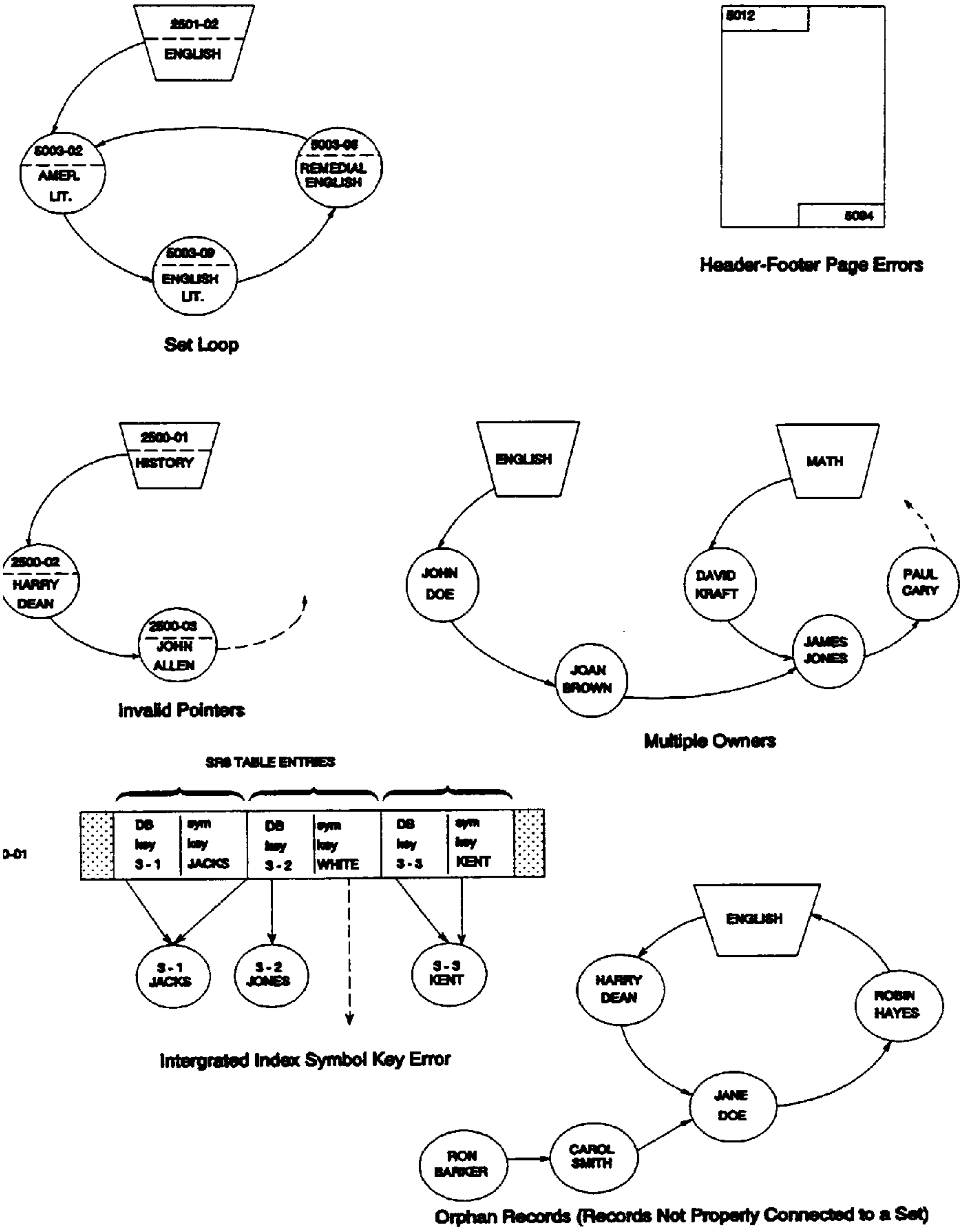


Figure 1.1: Examples of Physical Integrity Errors

1.4 When to Use CA-IDMS/DB Audit

CA-IDMS/DB Audit is useful both for periodic maintenance and for detecting and fixing errors that become apparent in special circumstances.

1.4.1 Periodic Maintenance

A database, like a car, will function more smoothly if it has periodic tune-ups. You can use CA-IDMS/DB Audit at regular intervals to examine the physical integrity of your database, to report the errors it finds, and to correct errors of physical integrity.

1.4.2 Special Situations

If your car is involved in an accident, or if it is not performing efficiently, it needs testing and repairs. Likewise, special situations indicate a need for examination of your database. Several conditions are likely to cause physical integrity errors in a database:

- Improper recovery from program, system, or hardware failures
- Malfunctions in the operation of a disk drive, disk controller, or other hardware
- Improper use of the CA-IDMS utility IDMSBCF, FIX PAGE function

If you know one of these conditions has occurred, you can use CA-IDMS/DB Audit to search for problems before running application programs.

Even if there has not been a known failure in the system, results of application programs sometimes indicate broken chains or other physical integrity errors. If an application program terminates abnormally or if the output from a program indicates broken chains, CA-IDMS/DB Audit can find and correct most of the physical integrity errors.

1.4.3 Logical Integrity

Once the physical integrity errors have been found and corrected by CA-IDMS/DB Audit, you can use other means to find and correct logical errors in your database. An example of a logical error is shown in the section on the Fix Phase in Chapter 2, “Concepts.”

1.5 Reports

CA-IDMS/DB Audit produces two types of reports: an Audit Report and an Integrity Analysis Report. The Audit Report displays a summary of the CA-IDMS/DB Audit run; the Integrity Analysis Report shows details of each physical integrity error, found and/or corrected. See Figure 1.2.

1.5.1 Audit Report

Every run of CA-IDMS/DB Audit produces an Audit Report. The report contains these sections:

- Parameter Validation
- Parameter Processing Option
- Areas, Records, and Sets
- Runtime Status and Statistics
- Audit Statistics
- Fix Statistics.

1.5.2 Integrity Analysis Report

Generation of the Integrity Analysis Report is controlled by the REPORTS parameter statement. You can generate reports for the current audit and fix operations; you can suppress the Integrity Analysis Report; or you can generate reports for a previous audit from which you saved the extract file.

The Integrity Analysis Report displays errors found in the audit phase and corrections (simulated or actual) made in the fix phase. The report is comprised of two sections:

Before Image Report--a report listing the details of all physical integrity errors found by CA-IDMS/DB Audit in the audit phase.

After Image Report--a report showing the corrections made to the database during the fix phase. The After Image Report displays either simulated or actual corrections, depending on whether the fix phase was run in simulation or update mode.

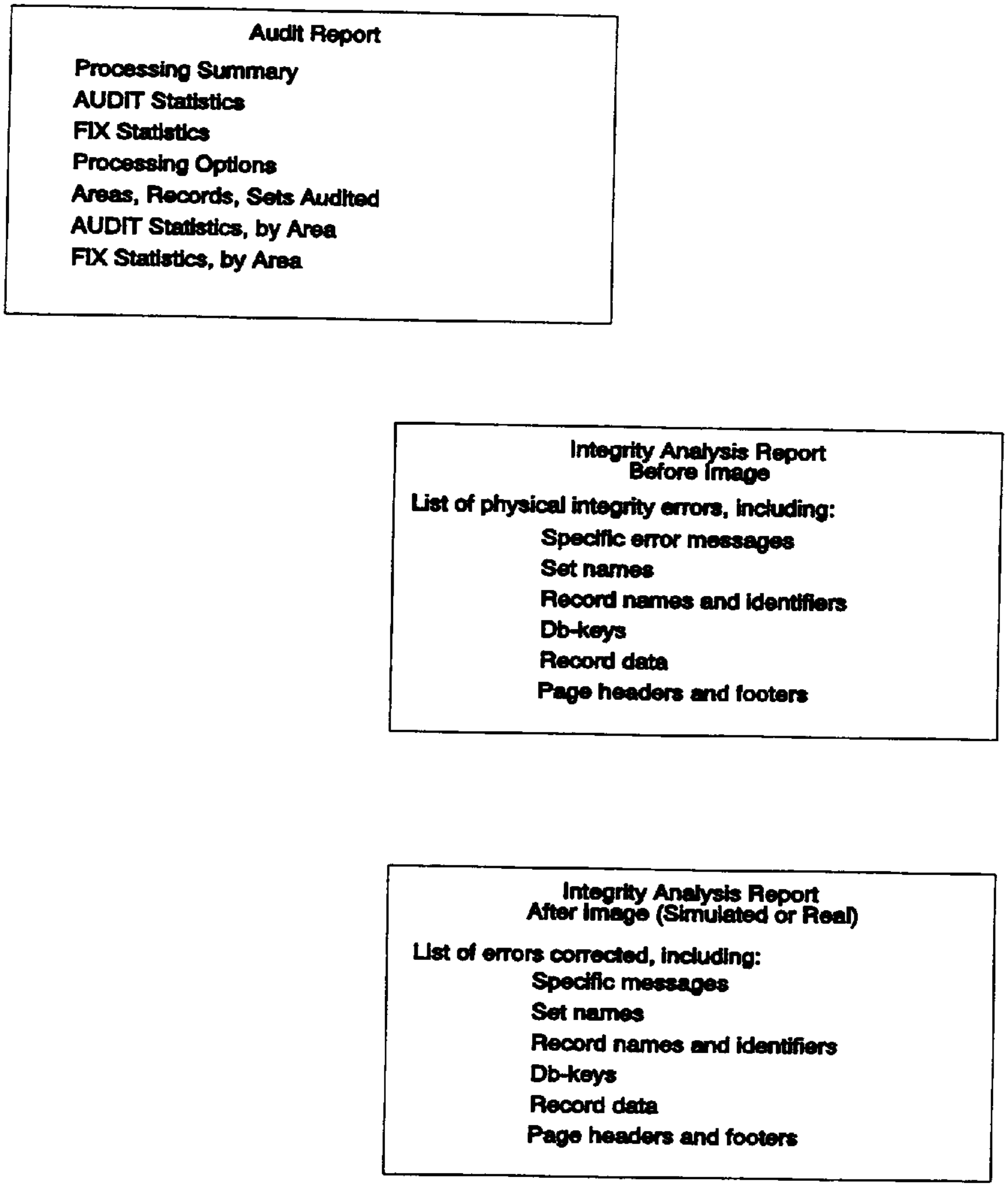
If CA-IDMS/DB Audit was run in the simulation mode, the report shows how the corrections would affect the database. After reviewing the simulated corrections, if you want to actually update the database, you can then make a separate fix run of CA-IDMS/DB Audit, using the extract file saved from the previous run. Updates should not be performed on the database between the audit phase and the fix phase of CA-IDMS/DB Audit, because such updates could alter the results or cause abnormal termination of the fix processing.

1.5.3 Report Formats

You can specify the type of notation to be used on the Integrity Analysis Report for display of data, and you can select a complete format or abbreviated format for display of sets in error.

Errors within sets can be displayed in only character notation (alphanumeric and special characters) or displayed in both character and hexadecimal notation.

Display of sets containing errors is available in either of two formats: complete format, which shows all of the members of each set in error; or abbreviated format, which shows the owner of the set in error, each member that is in error, and the next and prior members surrounding each member in error.



Audit Report

Processing Summary
AUDIT Statistics
FIX Statistics
Processing Options
Areas, Records, Sets Audited
AUDIT Statistics, by Area
FIX Statistics, by Area

Integrity Analysis Report
Before Image

List of physical integrity errors, including:

- Specific error messages
- Set names
- Record names and identifiers
- Db-keys
- Record data
- Page headers and footers

Integrity Analysis Report
After Image (Simulated or Real)

List of errors corrected, including:

- Specific messages
- Set names
- Record names and identifiers
- Db-keys
- Record data
- Page headers and footers

Figure 1.2: Reports

1.6 Control of CA-IDMS/DB Audit

Operation of CA-IDMS/DB Audit is controlled by parameter statements.

1.6.1 PROCESS Statement

The PROCESS parameter statement allows you to specify how the database is to be readied and which SORT messages to display on the Audit Report. Additional options initiate inclusive or exclusive processing, specify the audit type as either Standard or QuickCheck, and enable you to check your parameters before auditing begins.

1.6.2 AUDIT Statement

The AUDIT parameter statement allows you to control the audit phase of CA-IDMS/DB Audit. You can specify whether you want to find physical integrity errors in CALC sets, errors in user-defined sets, page errors, disconnections in sets, or index orphans (that is, records whose integrated index “up” pointer does not accurately point to the correct SR8 record). You can also specify any combination of those choices.

1.6.3 AREA, RECORD, and SET Statements

Associated with the AUDIT statement are AREA, RECORD, and SET statements, which allow you to limit the examination to a part of the database defined by the subschema.

1.6.4 FIX Statement

The FIX statement directs CA-IDMS/DB Audit to simulate corrections or to actually update the records in the database. The errors to be corrected must be recorded in an extract file, from either a current run or a previous run. If the audit was made in a previous run, you must be sure that the database was not updated between the audit run and the fix run.

Corrections can be made to page headers and footers; next, prior, and owner pointers in CALC sets, user-defined sets, and integrated index sets. Integrated index orphan records can be "adopted" and the orphan counts decreased. Logically deleted records (LDELs) can also be physically deleted from the database.

1.6.5 REPORTS Statement

The REPORTS statement controls production of the Integrity Analysis Report.

1.7 Unique Access to the Database

CA-IDMS/DB Audit uses a unique method to access the CA-IDMS database; it provides these efficient features:

- CA-IDMS/DB Audit can access the SR1 records and walk or fix the CALC set as it does any user defined record or set
- CA-IDMS/DB Audit can examine an entire physical page as well as an individual data record
- When CA-IDMS/DB Audit performs an area sweep it uses several buffers to reduce the amount of I/O wait time

1.8 Operating Considerations

In order to use CA-IDMS/DB Audit to best advantage, you should consider several operating procedures. Among these are backing up the database, selecting the readymode, selecting parameters that do not unduly lengthen the run time, and reviewing the logical integrity.

1.8.1 Back Up the Database

CA-IDMS/DB Audit does not write before images and after images of changed data to a journal file. Therefore, you should back up the database before attempting fix processing. You should also back up the database after fix processing has been performed.

1.8.2 Plan Space Allocation

If you are auditing large sets, you can allocate storage for the processing tables to disk instead of core.

1.8.3 Select Readymode

If you are using CA-IDMS/DB Audit as a periodic maintenance tool, you can select RETRIEVAL mode. The database is readied in SHARED RETRIEVAL mode, and CA-IDMS/DB Audit can share the database with other programs. In this mode, however, CA-IDMS/DB Audit can only simulate corrections; it cannot actually update the database in RETRIEVAL mode. Also, if you use RETRIEVAL mode, you should ensure that no other program updates the database while CA-IDMS/DB Audit is auditing or fixing the database.

1.8.4 Plan Run Time

The total amount of time to run CA-IDMS/DB Audit depends on which parameters you specify. If the utility is required to access record occurrences and set occurrences over and over, the operating time is significantly increased. The most time consuming features are DISCONNECT and WALKNEXT. You can run CA-IDMS/DB Audit without these parameters, and then use them in another run if there are indications on the Audit Report that there are disconnected or orphan records. The Standard audit type is generally slower than the QuickCheck type and QuickCheck provides the added bonus of orphan detection without WALKNEXT.

1.8.5 Review Logical Integrity

Review the reports to understand the errors that were detected and the way the errors have been fixed. It may be necessary to validate the logical integrity of the corrected set occurrences. CA-IDMS/DB Audit finds and corrects only physical integrity errors.

1.9 Environment

CA-IDMS/DB Audit supports CA-IDMS Release 15.0; it runs under OS/390, VSE/ESA, and VM/ESA.

1.10 CA-IDMS/DB Audit Release 15.0 Features

CA-IDMS/DB Audit now uses the following features of CA-IDMS database I/O:

- XA database buffers and control blocks
- ESA dataspace support
- Dynamic database file allocation
- Unrestricted SEGMENT name usage as database names

For more information on these CA-IDMS database features, see *CA-IDMS Database Administration* and *CA-IDMS System Operations*.

Additionally, features of the CA-IDMS database engine and the SYSIDMS PREFETCH parameter for read-ahead processing replace the read-ahead processing previously provided by FASTSCAN and EXCP I/O level processing previously defined by the GSDTPARM installation defaults.

Chapter 2. Concepts

CA-IDMS/DB Audit detects a number of different physical integrity errors. The first section of this chapter describes each type of error and provides an illustrated example. The conventions and the data structure diagram used for these examples are presented on the following pages.

QuickCheck and Standard are the two auditing methods CA-IDMS/DB Audit provides. The method you select determines the types of errors CA-IDMS/DB Audit can locate. The auditing method also greatly affects processing time.

In the audit phase, CA-IDMS/DB Audit uses a variety of processes to find errors. This section of the chapter describes each process, the parameter used to initiate each process, and the specific error type it searches for. Set diagrams illustrate these processes.

In the fix phase, CA-IDMS/DB Audit fixes most error types located by the audit phase. This section describes which errors can be fixed and also presents diagrammed examples.

2.1 Conventions

The diagrams in this chapter use the following conventions.

- Symbols--Figure 2.1 presents the symbols used in the diagrams for record types and db-keys.

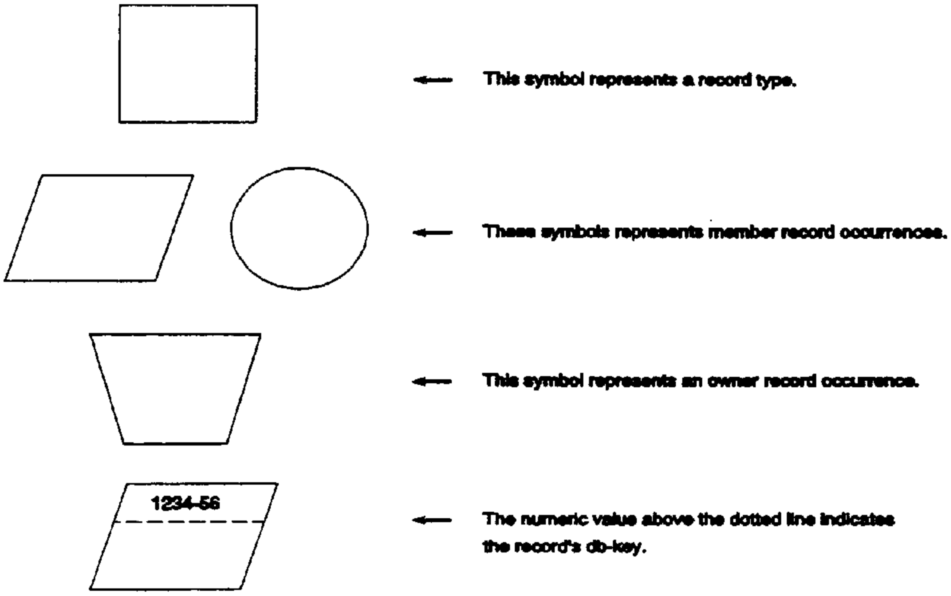
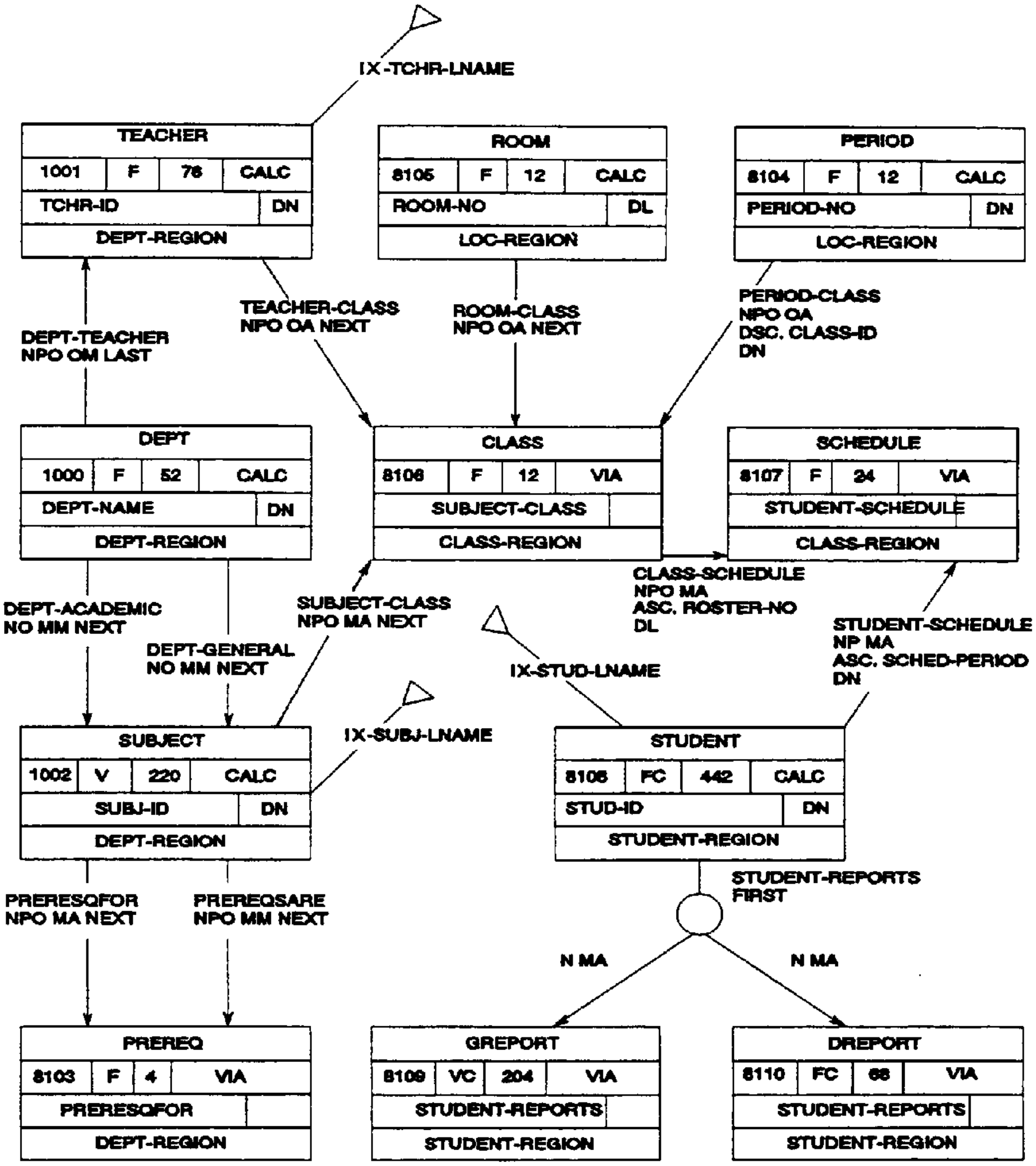

This symbol represents a record type.
These symbols represents member record occurrences.
This symbol represents an owner record occurrence.
1234-56
The numeric value above the dotted line indicates the record's db-key.

Figure 2.1: Record Symbols

- Pointers--For simplicity, most diagrams in this chapter illustrate only the next pointer and not the prior or owner pointers. Any examples with additional pointers are labeled accordingly.
- Terminology--The words record and set are used to mean record occurrence and set occurrence.
- Lettercase--The text of this chapter presents owner record occurrences in uppercase. Example: ENGLISH. The text of this chapter capitalizes member record occurrences. Example: English Literature.

2.2 Data Structure Diagram

Figure 2.2 presents the data structure diagram TESTSUB, the sample database used in the examples in this chapter.



NOTE: All IX sets are system-owned integrated index sets residing in INDEX-REGION.

Figure 2.2: Data Structure Diagram

2.3 Integrity Errors

CA-IDMS/DB Audit can detect the following categories of physical integrity errors when auditing a database:

- Pointer Errors
- Set Errors
- Integrated Index Errors
- Userset Orphan Record Occurrences
- Fragmented (variable length) Record Errors
- Compressed Record Errors
- Page Errors

All physical integrity errors are reported on the Integrity Analysis Report. For each error encountered, the report lists the members of the set, plus an error message just below the error. If SETMBRS=ALL (default) is specified, every member of the set is listed. If SETMBRS=ERRSONLY is specified, the only records displayed are the owner, the member in error, and the prior and next records for each member in error.

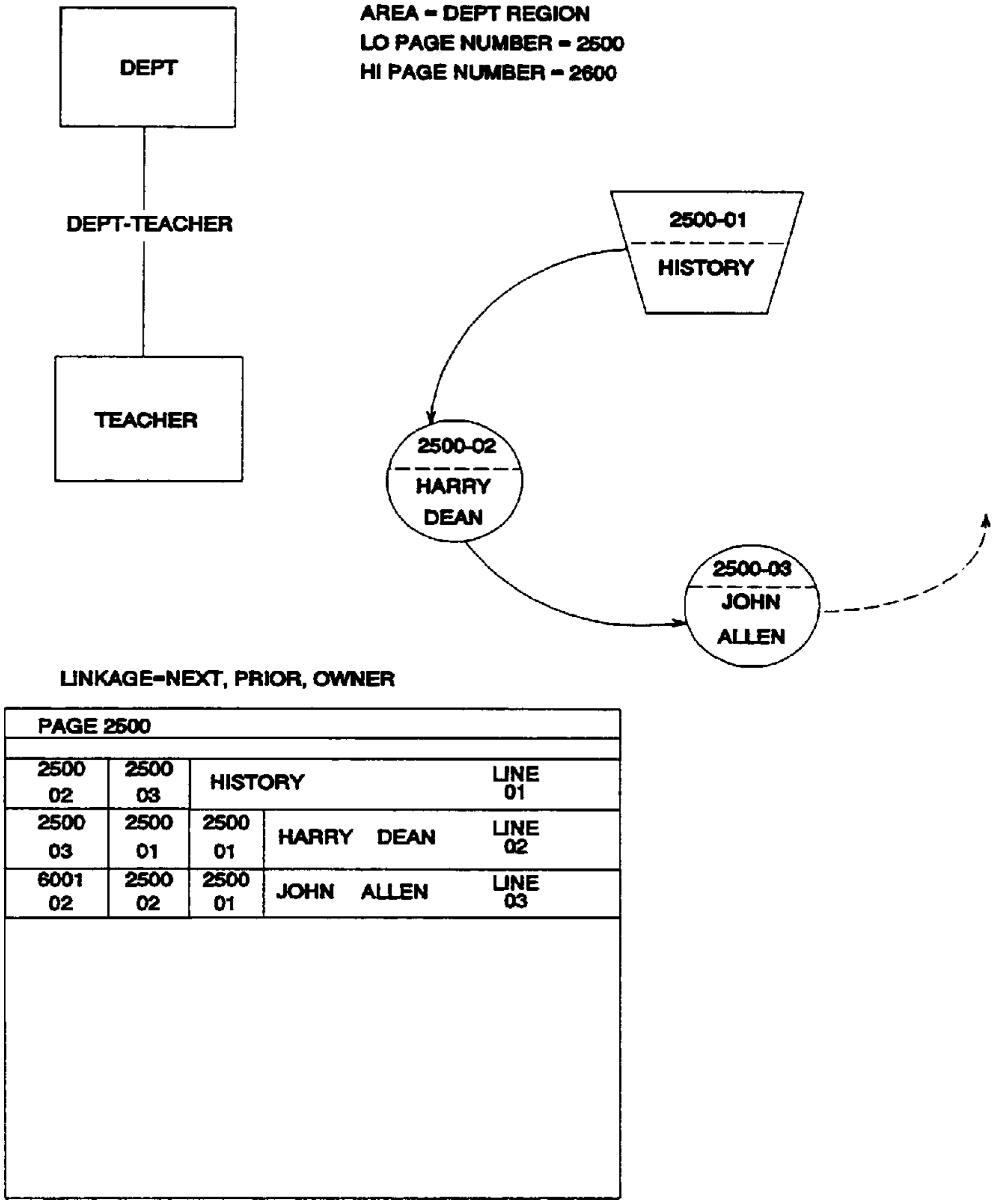
2.3.1 Pointer Errors

Pointer errors are related to an individual record occurrence in a specific set. CA-IDMS/DB Audit can correct pointer errors in user-defined sets, integrated index sets, and CALC sets. There are six types of pointer errors.

1. Next, prior, or owner db-key points outside area page ranges in the subschema

CA-IDMS/DB Audit verifies that all member records of a multi-member set to be audited, and the areas where the records reside, are copied into the subschema used by CA-IDMS/DB Audit. (Except CA-IDMS/DB Audit cannot verify that all CALC records are copied into the subschema when CALCSET audit is requested.) Therefore, the page number for a next, prior, or owner pointer must be for a page within an area copied into the subschema.

For example, in Figure 2.3, assume that the only area copied in the subschema is DEPT-REGION. As a result, only records within the DEPT-REGION area can be copied into the subschema. In this case, John Allen's next db-key is not within the DEPT-REGION page ranges and is in error. CA-IDMS/DB Audit will fix the error by correcting the pointer so that it points to HISTORY.



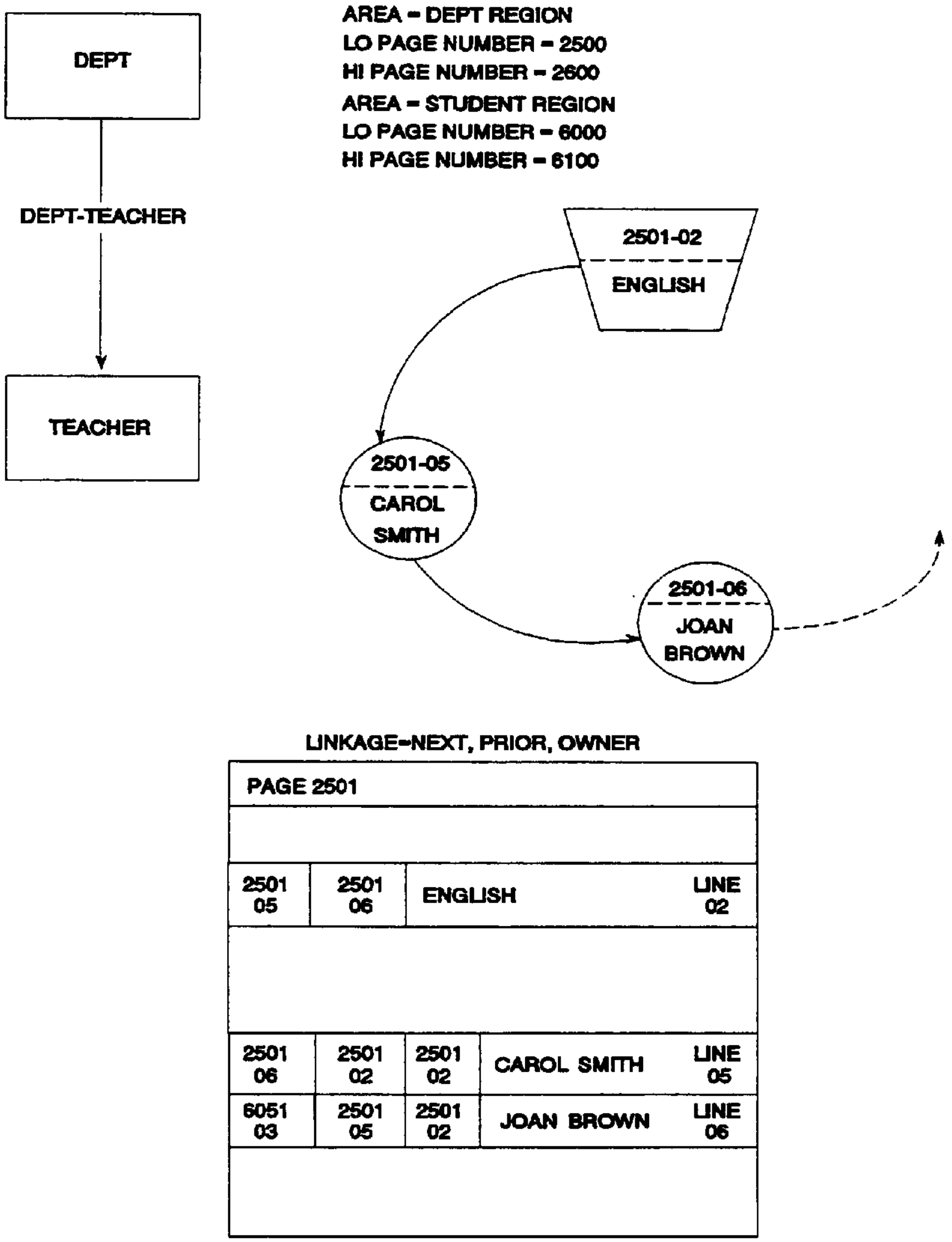
PAGE 2500				
2500 02	2500 03	HISTORY		LINE 01
2500 03	2500 01	2500 01	HARRY DEAN	LINE 02
6001 02	2500 02	2500 01	JOHN ALLEN	LINE 03

Figure 2.3: Area Points to Area Outside Page Range

2. Next, prior or owner db-key points to a non-readied area

CA-IDMS/DB Audit verifies whether all member records of a set to be audited, and the areas where the records reside, are copied into the subschema used by CA-IDMS/DB Audit. (Except CA-IDMS/DB Audit cannot verify that all CALC records are copied into the subschema when CALCSET audit is requested.) Therefore, the page number in a next, prior, or owner pointer must be for a page within an area readied by CA-IDMS/DB Audit.

For example, in Figure 2.4, assume that the DEPT-REGION area was readied and the STUDENT-REGION area was not readied, because NOACCESS was specified by the user. As a result, Joan Brown's next db-key is invalid, because it points to a record that is in the non-readied area. CA-IDMS/DB Audit corrects the next db-key to point to ENGLISH.



PAGE 2501				
2501 05	2501 06	ENGLISH		LINE 02
2501 06	2501 02	2501 02	CAROL SMITH	LINE 05
6051 03	2501 05	2501 02	JOAN BROWN	LINE 06

Figure 2.4: Record Points to Non-Readied Area

3. Next, prior, or owner db-key points to a non-existent record

In Figure 2.5, Carol Smith's next db-key points within the DEPT-REGION area, but the record for the specified db-key does not exist. Because the set is a user set, CA-IDMS/DB Audit corrects the next pointer so that it points to the Joan Brown record.

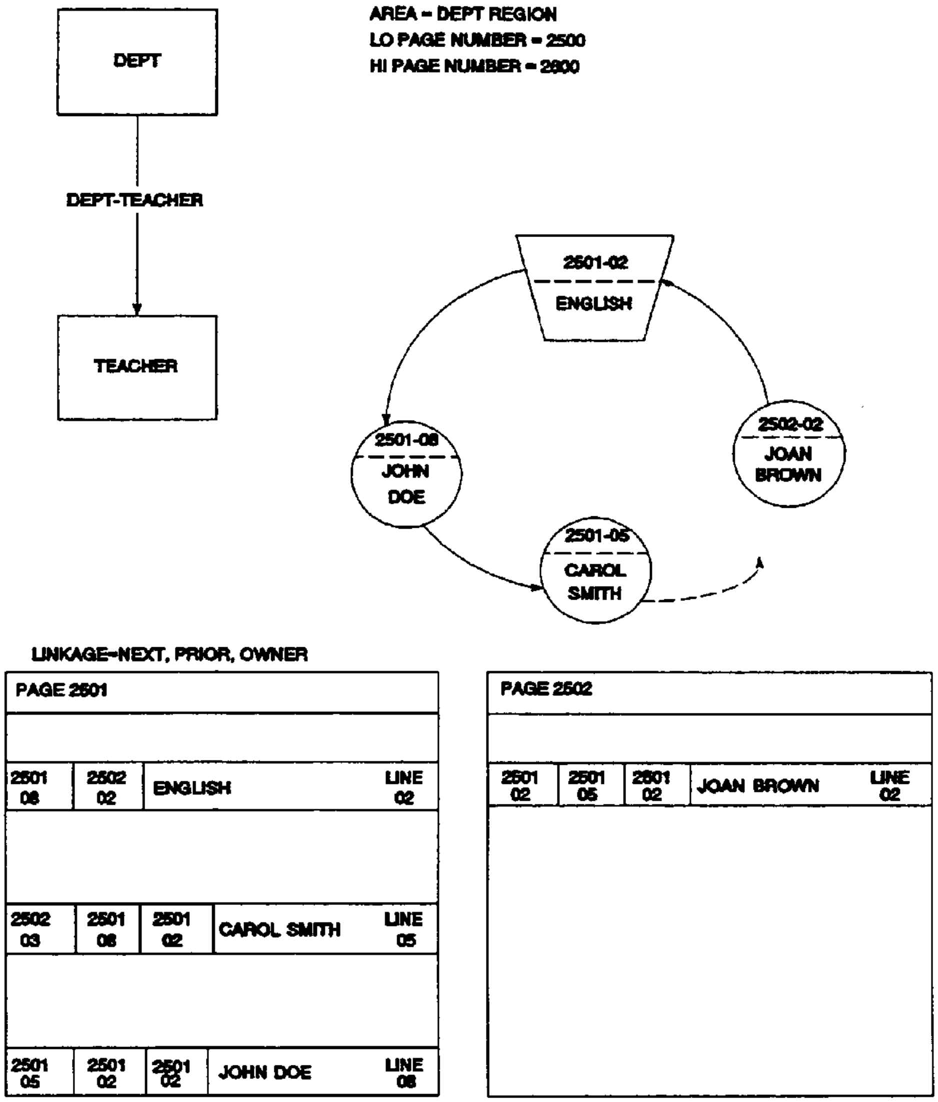


Figure 2.5: Record Points to Non-Existent Record

4. Next, prior or owner db-key points to an invalid record type for the set

CA-IDMS/DB Audit verifies whether all member records of a set are copied into the subschema used by CA-IDMS/DB Audit. (Except CA-IDMS/DB Audit cannot verify that all CALC records are copied into the subschema when CALCSET audit is requested.) Therefore, a valid next, prior, or owner db-key must point to a valid record type for that set.

For example, in Figure 2.6, the only valid record types for the DEPT-TEACHER set are a DEPT record and a TEACHER record. Therefore, Jane Doe's next db-key is in error because it is pointing to SUBJECT record Creative Writing. CA-IDMS/DB Audit corrects the error by changing the next pointer so that it points to the MATH record.

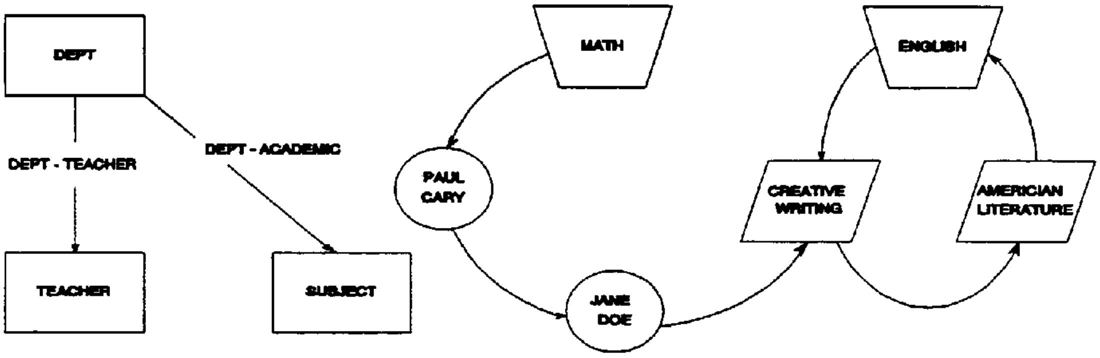


Figure 2.6: Record Points to Invalid Record Type

5. Next, prior or owner db-key is null

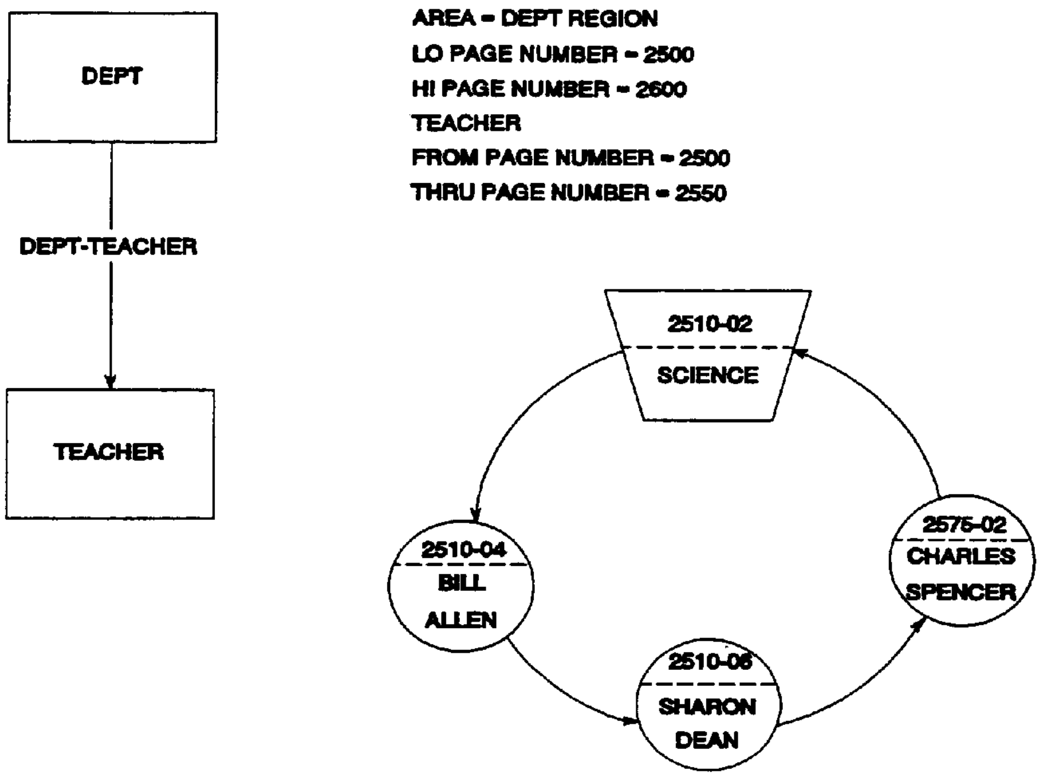
The next, prior, or owner db-key for this record is equal to HIGH-VALUES. If the set is user-defined, CA-IDMS/DB Audit corrects the error by entering a valid pointer.

6. Db-key of record is outside the from/thru page range for record type

This type of error probably occurs because a schema was compiled when the database was originally developed, and a FROM/THRU page range was not specified for the record type. As a result, the FROM/THRU page range defaulted to the low-area-page/high-area-page number of the area. The schema was recompiled with a different FROM/THRU page range for the record type. Some records stored before the recompilation now fall outside the new FROM/THRU page range.

This error does not affect the walking of a set, but it can cause CALC records stored before the recompilation to be lost. This loss happens because the CA-IDMS CALC routine uses the current FROM/THRU page range in its calculation of the randomized CALC page number.

In Figure 2.7, the FROM/THRU page range for the TEACHER record is pages 2500 thru 2550. The db-key of Charles Spencer falls outside this range. CA-IDMS/DB Audit does not fix this type of error.



LINKAGE=NEXT, PRIOR, OWNER

PAGE 2510

2510 04	2575 02	SCIENCE		LINE 02
2510 06	2510 02	2510 02	BILL ALLEN	LINE 04
2575 02	2510 04	2510 02	SHARON DEAN	LINE 06

PAGE 2575

2510 02	2510 06	2510 02	CHARLES SPENCER	LINE 02

Figure 2.7: Record Points to Db-key Outside Page Range

2.3.2 Set Errors

Set errors are related to a set occurrence or to a group of record occurrences for a specific set. There are five major groups of set errors:

- Disconnected Record
- Sorted Sets
- Multiple Owner Records
- Invalid Db-keys
- CALC Sets

2.3.2.1 Disconnected Record

There are two types of disconnected records:

1. A fully disconnected record

A fully disconnected record is one that is not connected to any set occurrence. All of the disconnected record's db-key values pointing to the specified set are HIGH-VALUES. CA-IDMS/DB Audit finds disconnected records when the AUDIT statement indicates DISCONNECT=EXPLICIT (plus the DISCONNECT option in RECORD or SET statements), or DISCONNECT=ALL.

Mandatory-automatic sets are searched for disconnections when you specify the ALL or the EXPLICIT option. Because CA-IDMS automatically connects mandatory-automatic sets at store time, and allows disconnect only at ERASE, a record belonging to a mandatory automatic set should never appear disconnected. In Figure 2.8, David Herman is an example of a fully disconnected record because it is not connected to any set.

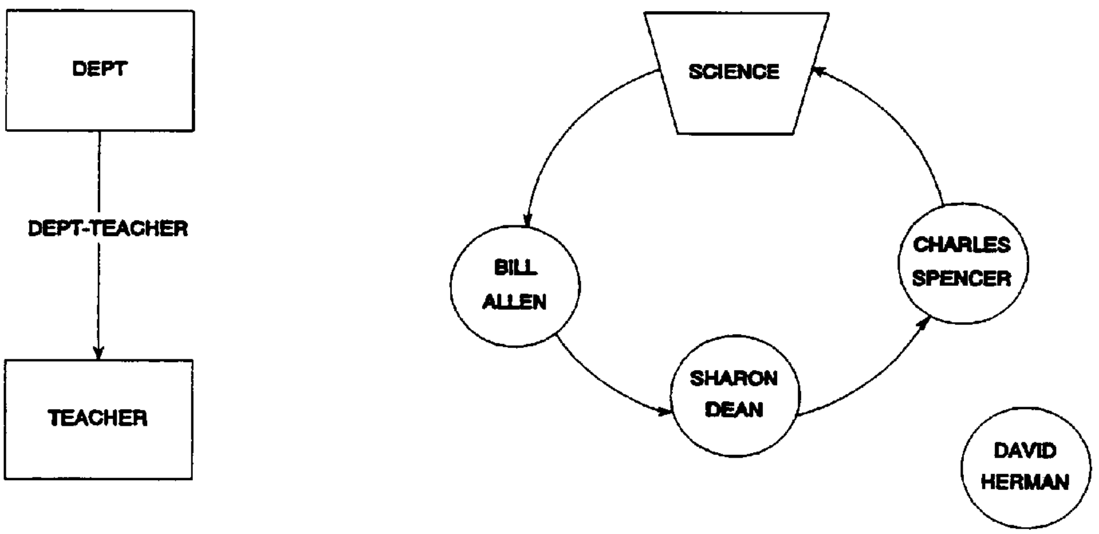


Figure 2.8: Record Fully Disconnected from Set

2. A partially disconnected record

A partially disconnected record is a record that points into a set, but no record in the set points to it. Figure 2.9 presents a partially disconnected record.

Partially disconnected records are also referred to as orphan records. Orphans are described in greater detail later in this chapter.

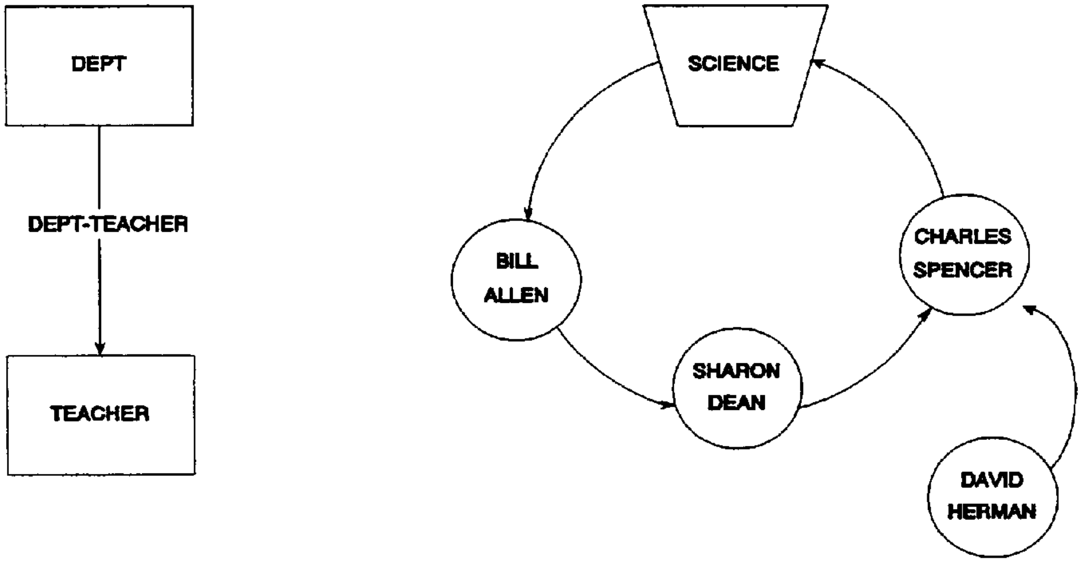


Figure 2.9: Record Partially Disconnected from Set

2.3.2.2 Sorted Sets

There are two types of errors in sorted sets:

1. Sorted set is out of sequence

In a set that is sorted in descending order, the sort key of the previous record in the set is greater than the sort key of the current record. In a set that is sorted in ascending order, the sort key of the previous record is less than the sort key of the current record. For example, in Figure 2.10, the DEPT-TEACHER set should be sorted in ascending order by last-name. Jane Doe is out of sequence, however; she should precede Harold Jones in the set. CA-IDMS/DB Audit corrects this set so that ENGLISH points to Jane Doe, Jane Doe points to Harold Jones, and Harold Jones points to Carol Smith.

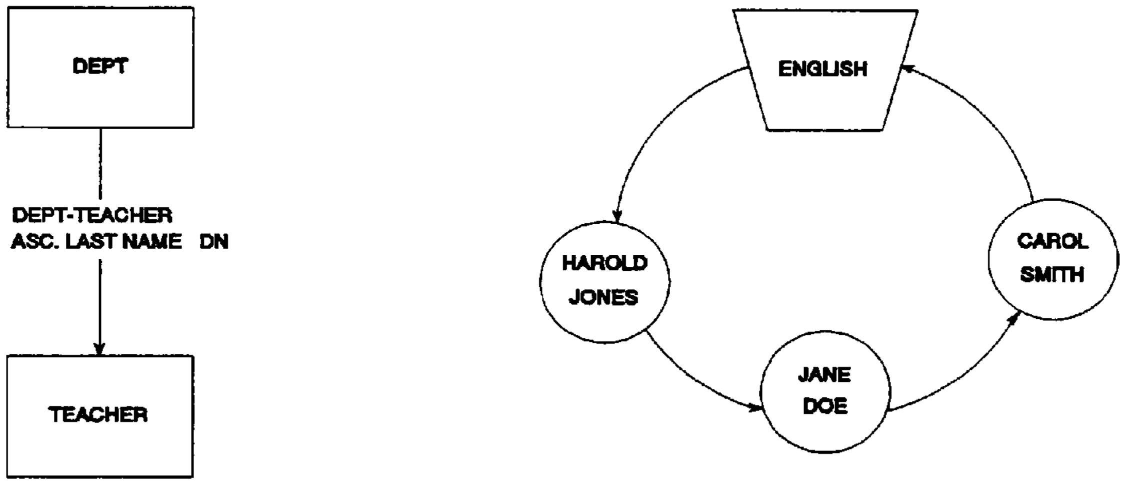


Figure 2.10: Sorted Set Out of Sequence

2. Duplicates in a sorted set when duplicates are not allowed

The sort key of the current record equals the sort key of the previous record in the set occurrence. For example, in Figure 2.11, duplicates are not allowed within the DEPT-TEACHER set. David Herman, however, is found to be in the set twice. CA-IDMS/DB Audit does not fix this type of error.

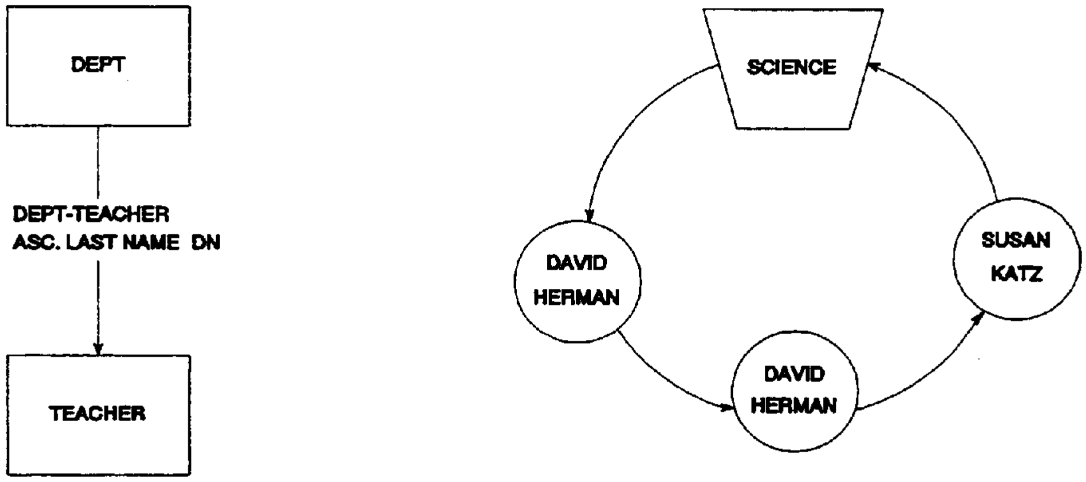


Figure 2.11: Sorted Set with Duplicates

2.3.2.3 Multiple Owner Records

There are two error situations that cause CA-IDMS/DB Audit to report a second owner in a set. Only Standard auditing detects multiple owners in a set. The way CA-IDMS/DB Audit corrects multiple owner record errors is described in detail later in this chapter.

1. A single set contains two owners

In Figure 2.12, the set occurrence owned by the ENGLISH record is a complete set chain. As the set is walked from the ENGLISH record, however, another owner record (MATH) is encountered.

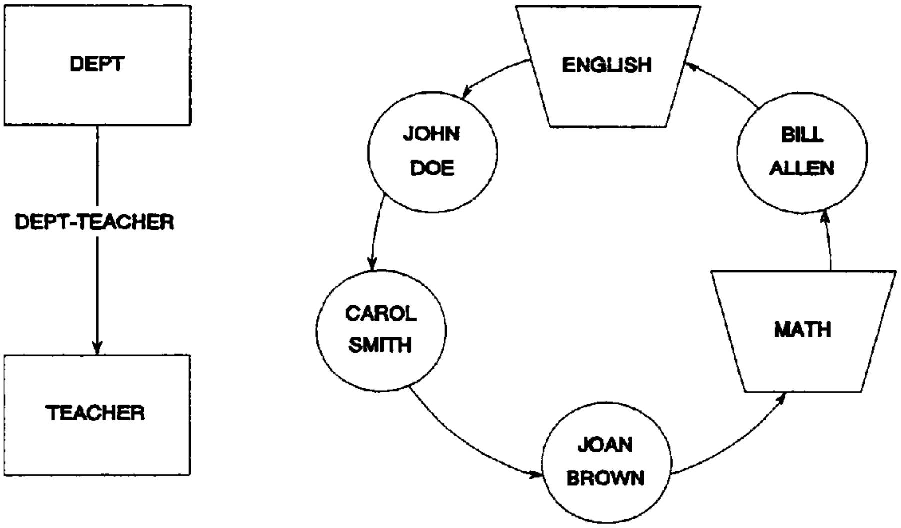


Figure 2.12: One Set Contains Two Owners

2. A broken set points into a complete set

The set owned by the HISTORY record in Figure 2.13 is broken. When this set is walked in the next direction, a second owner record (CHEMISTRY) is retrieved. Carol Smith's next db-key points to James Lyons' record, which is in the set occurrence owned by CHEMISTRY.

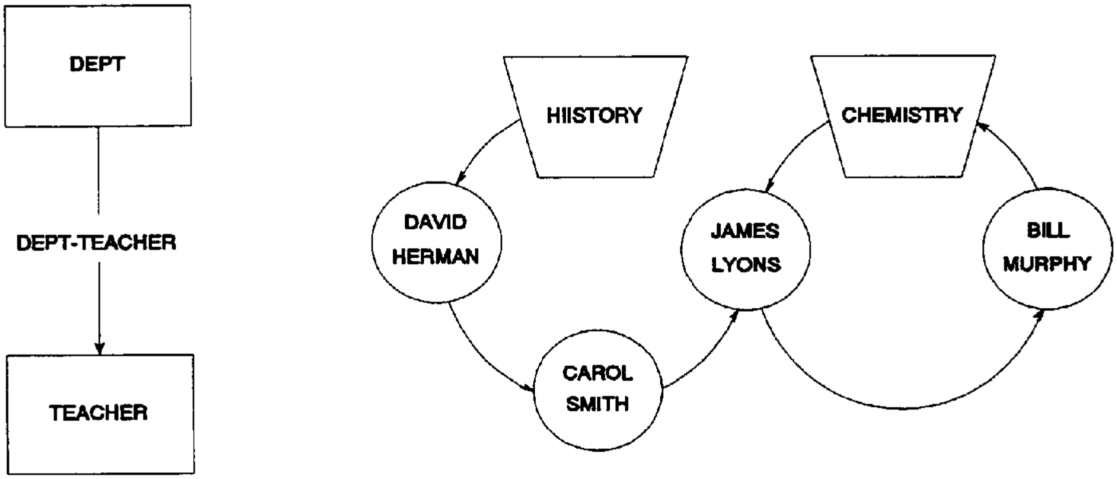


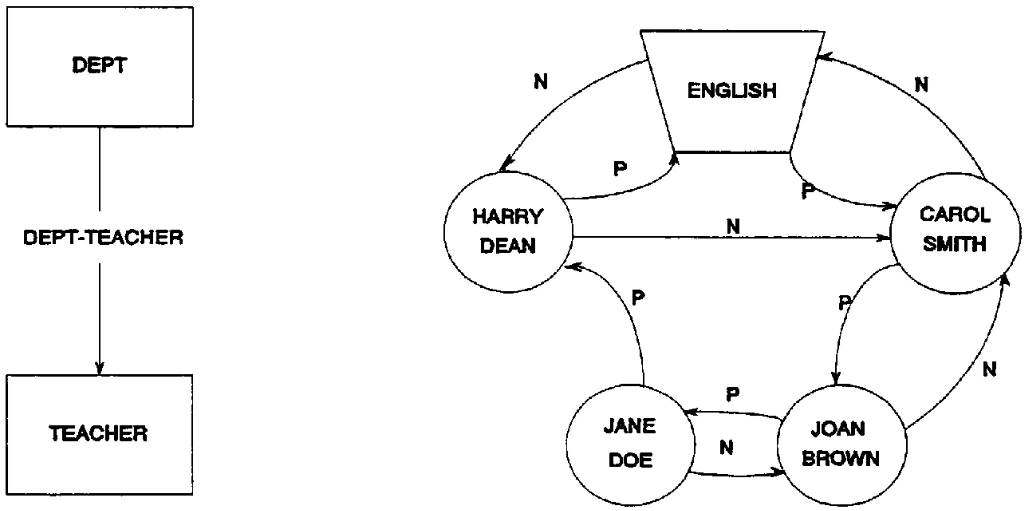
Figure 2.13: Broken Set Points to a Complete Set

2.3.2.4 Invalid Db-keys

CA-IDMS/DB Audit reports three types of invalid db-keys:

1. Prior db-key does not point to the prior member of set

When walking a set in the next direction, the prior db-key does not point to the prior record in the set. For example, in Figure 2.14, walking next from the ENGLISH record, Harry Dean is retrieved. Walking next again, Carol Smith is retrieved but her prior db-key does <u>not</u> point back to Harry Dean; it points to another valid record occurrence (Joan Brown).



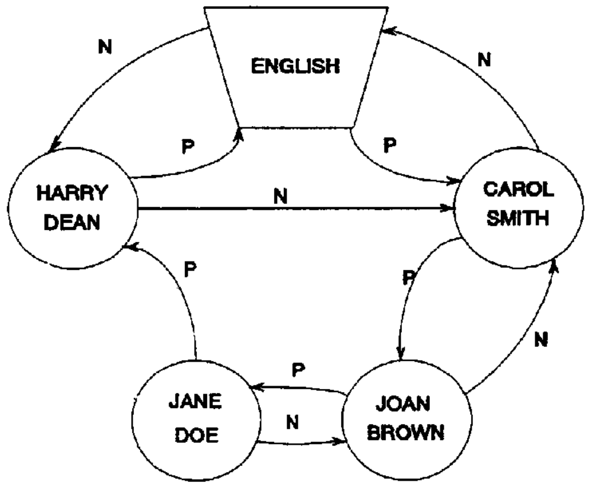
Carol Smith's PRIOR db-key does not point back to Harry Dean

Figure 2.14: Prior Pointer Does Not Point to Prior Record

2. Next db-key does not point to the next member of set

When walking a set in the prior direction, the next db-key does not point to the next record in the set (the record just walked from). For example, in Figure 2.15, walking prior from the ENGLISH record, all next db-keys point back to the previous record until Harry Dean is retrieved. Harry Dean's next db-key does not point to Jane Doe, but to Carol Smith.

In both of these situations, CA-IDMS/DB Audit evaluates the error by walking the set in the next and prior direction and fixes the set such that Harry Dean's next db-key points to Jane Doe.



When walking PRIOR, Harry Dean's NEXT db-key does not point to Jane Doe

Figure 2.15: Next Pointer Does Not Point to Next Record

3. Owner db-key does not point to the owner of the set

In Figure 2.16, Jane Doe's owner db-key does not point to the owner of the DEPT-TEACHER set. Instead it points to the HISTORY record, which is an owner record of another DEPT-TEACHER set occurrence.

CA-IDMS/DB Audit corrects this error by making Jane Doe's owner db-key point to English.

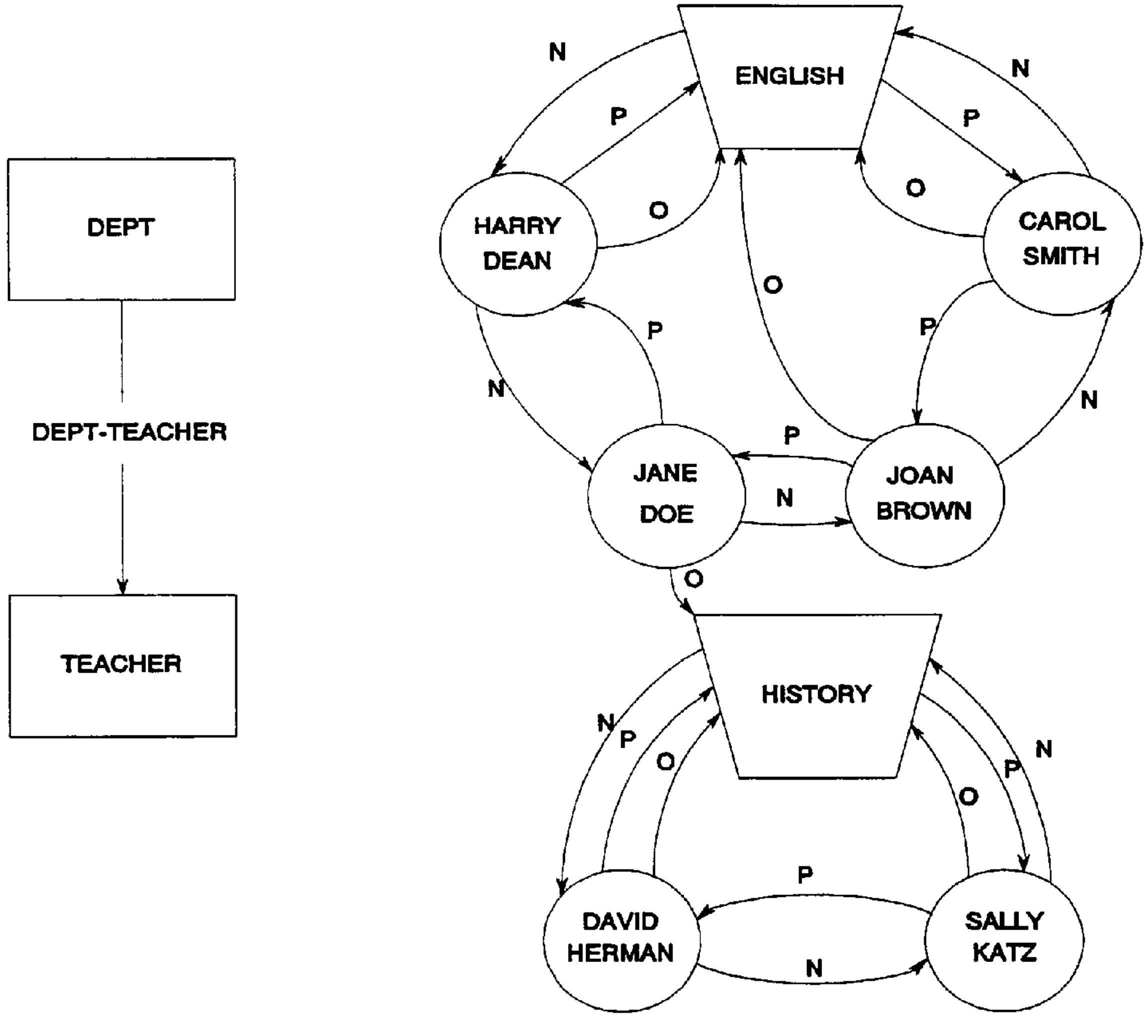


Figure 2.16: Owner Db-key Does Not Point to Set Owner

2.3.2.5 CALC Sets

To identify any errors in the CALC set, CALCSETS must be specified on the AUDIT statement. There are four types of CALC set errors:

1. The CALC set is out of sequence

The sequence of the CALC set is determined by the same rules that govern any multi-member sorted set. This error is fully described in the earlier section on Sorted Sets.

2. A VIA or DIRECT record is connected into the CALC set

This error is fully described in the earlier section Pointer Errors. Next, prior, or owner db-key points to an invalid record type for the set.

3. A CALC record is disconnected from the CALC set

This error is fully described in the earlier section or Disconnected Record.

4. A CALC record is connected to the CALC set on an incorrect page

Every page of an area contains a system record (SR1) that is the owner of all CALC records whose keys randomize to that page. The CALC set is the only set for which CA-IDMS/DB Audit can determine the accuracy of the set owner by invoking the IDMSCALC (or IDMSCLCX) routine for each CALC record. CA-IDMS/DB Audit ensures that each CALC record is connected to the correct CALC set. (If a CALC record is not connected to the correct CALC set, it cannot be located by an OBTAIN CALC command.) In Figure 2.17, ART DEPARTMENT is an example of a CALC record connected to the CALC set on an incorrect page. CA-IDMS/DB Audit corrects this so that ART DEPARTMENT is connected to the CALC set on page 88,004.

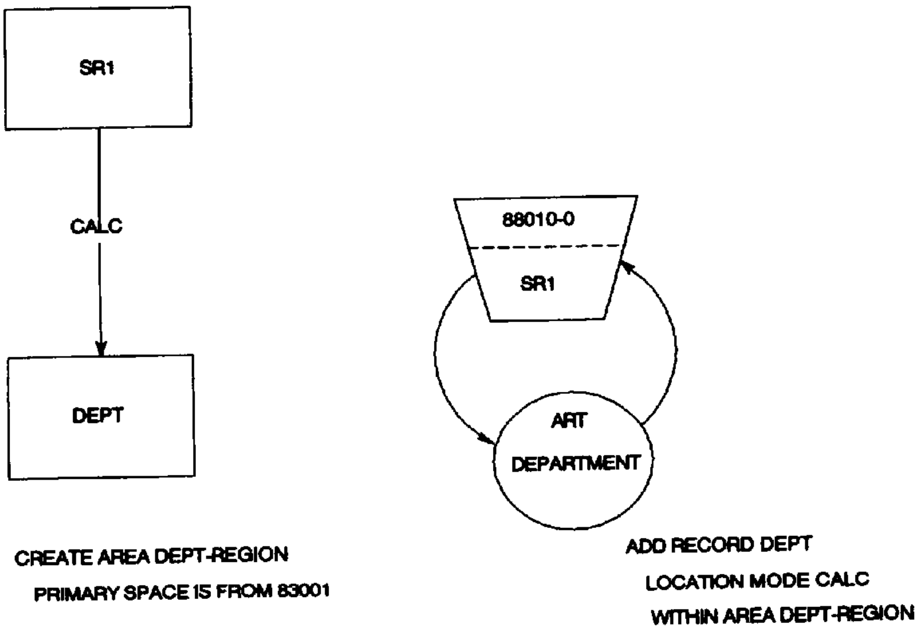
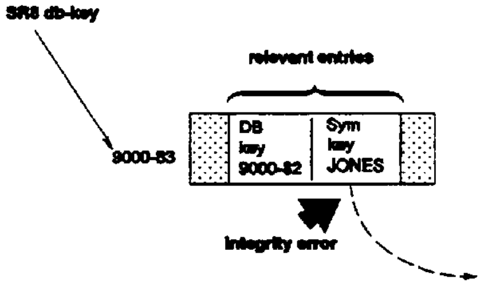


Figure 2.17: CALC Record Connected to an Incorrect CALC Set

2.3.3 Integrated Index Integrity Errors

To detect integrated index errors, specify INDEX in the AUDIT parameter statement. In the diagrams of integrated index sets, each SR8 record is shown with its db-key and entries relevant to the illustrated physical integrity error. Entries not shown in the SR8 record are represented by the shaded portion. A thick black arrow shows the location of an integrity error.



There are three major groups of integrated index errors:

- Pointer errors in the SR7/SR8 structure
- Integrated index orphans and orphan count
- Miscellaneous integrated index errors

2.3.3.1 Pointer Errors in Integrated Index SR7/SR8 Structure

Errors in next, prior, and owner pointers in integrated index sets are detected when you specify USERSETS in the AUDIT statement, and are fixed when you specify USERSETS in the FIX statement. CA-IDMS/DB Audit generates the same messages for pointer errors in other sets in the database. Figure 2.18 shows a typical integrated index structure with a bad next pointer. The NEXT pointer in record 9000-03 points to the level above instead of the next record.

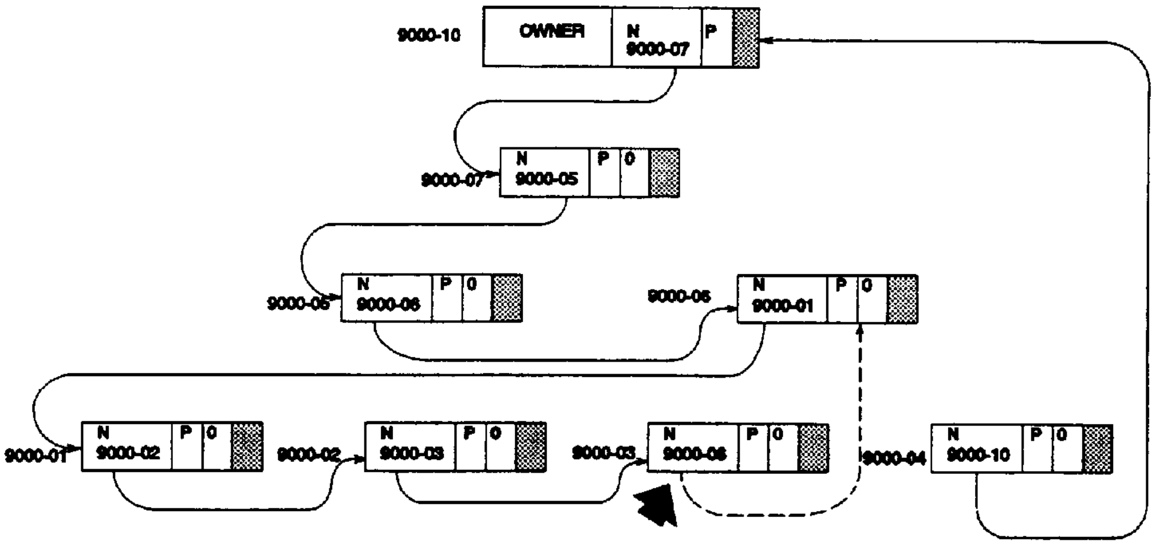


Figure 2.18: Next Pointer Error in Integrated Index Set

2.3.3.2 Orphans Records in Integrated Index Sets

CA-IDMS/DB Audit detects integrated index orphan records. An integrated index orphan occurs when the upper level pointer of an SR8 record or a member record of an integrated index set does not point to the correct record in the level above it. Integrated index orphans are not integrity errors because this condition can occur normally when the database spawns new SR8 records to accommodate the addition of new member records. Their presence, however, can degrade performance.

For example, in Figure 2.19 upper level record 9000-06 points to record 9000-03 but record 9000-03 points back to record 9000-05. To fix this condition, specify INDEXORPHANS in the FIX parameter statement. INDEXORPHANS also causes the orphan counts in SR8 records to be corrected.

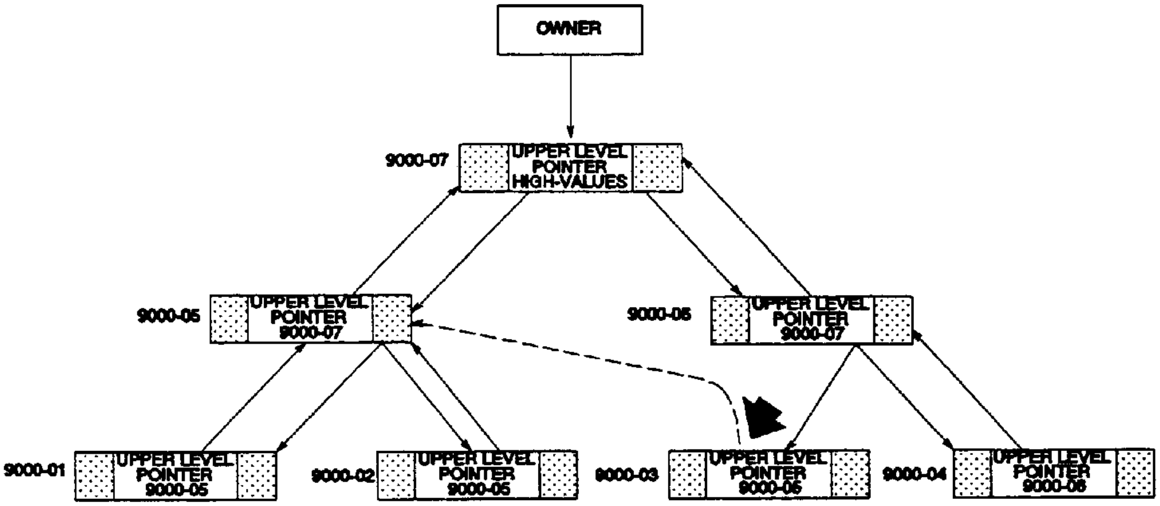

OWNER
9000-07
UPPER LEVEL POINTER HIGH-VALUES
9000-05
UPPER LEVEL POINTER 9000-07
9000-06
UPPER LEVEL POINTER 9000-07
9000-01
UPPER LEVEL POINTER 9000-05
9000-02
UPPER LEVEL POINTER 9000-05
9000-03
UPPER LEVEL POINTER 9000-06
9000-04
UPPER LEVEL POINTER 9000-06

Figure 2.19: Orphan Record in Integrated Index Set

2.3.3.3 Miscellaneous Integrated Index Errors

CA-IDMS/DB Audit can detect, but not fix, seven other types of integrity errors in an integrated index.

1. Index entry db-key errors detected in integrated index records

When checking the db-keys of records in an integrated index set, CA-IDMS/DB Audit either could not find the record pointed to by the db-key, or found errors within the record.

- **Record not found**. The record indicated by the db-key could not be found. For example, in Figure 2.20, the bottom level SR8 record points to a member record 6-7 that cannot be found.
- **Record type incorrect.** The db-key points to a record whose type does not belong in the set. For example, the db-key in a TEACHER index set points to a CLASS record.
- **Record not within page range**. The page where the SR8 record resides is not within the designated page range. The page range for an SR8 is determined by the high and low pages of the set to which it belongs. These pages can be defined in the schema. If the pages are not specifically defined, the page range is the same as the page range for the owner record.
- **Record located in non-readied area**. An SR8 record points to a record that is in a non-readied area.

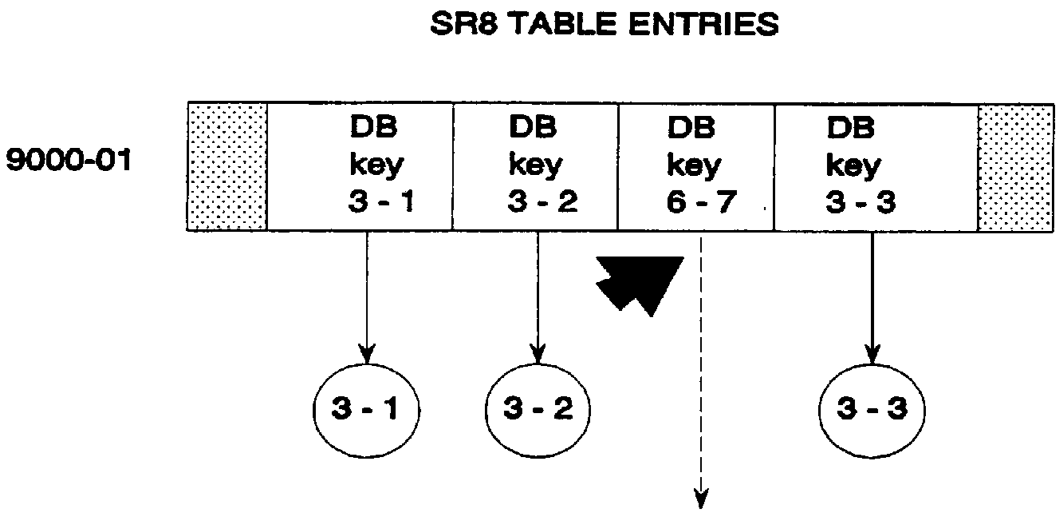


Figure 2.20: Integrated Index Entry Not Found

2. SR8 upper level pointer is not high values

In a top level SR8 record, the upper level pointer should be null (equal to HIGH-VALUES). If the upper level pointer in a top level SR8 is not null, CA-IDMS/DB Audit lists the error under db-key errors and indicates that the upper level pointer is not HIGH-VALUES.

In the example in Figure 2.21 the upper level pointer in record 9000-07 points to a record in another set.

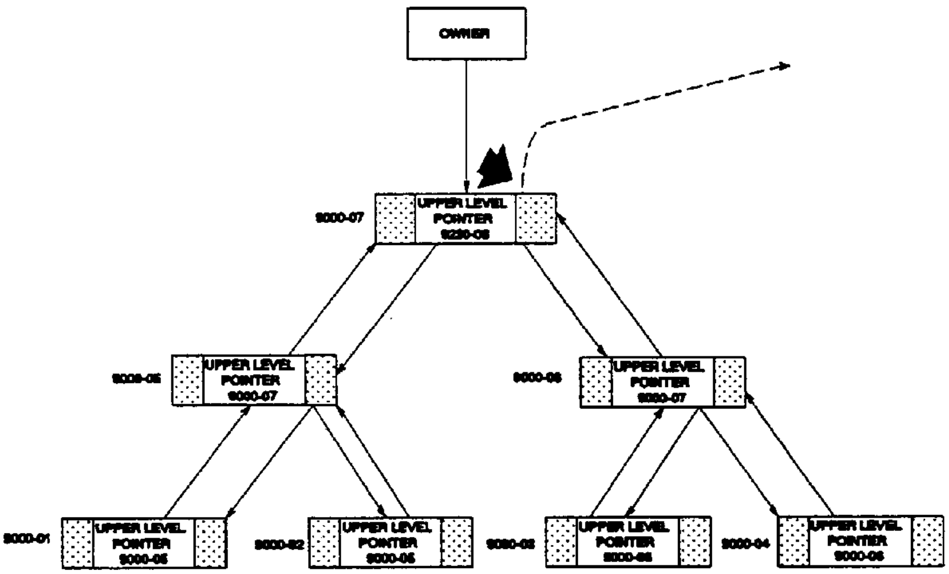

OWNER
9000-07
UPPER LEVEL POINTER 9230-08
9000-05
UPPER LEVEL POINTER 9000-07
9000-06
UPPER LEVEL POINTER 9000-07
9000-01
UPPER LEVEL POINTER 9000-05
9000-02
UPPER LEVEL POINTER 9000-05
9000-03
UPPER LEVEL POINTER 9000-06
9000-04
UPPER LEVEL POINTER 9000-06

Figure 2.21: Integrated Index Invalid Upper Level Pointer

3. Level sequence errors detected in integrated indexes

The level number of an SR8 record in an integrated index set is in error. For example, in Figure 2.22 the level number in Level 1 should be 1, but the records contain Level 0 and 3.

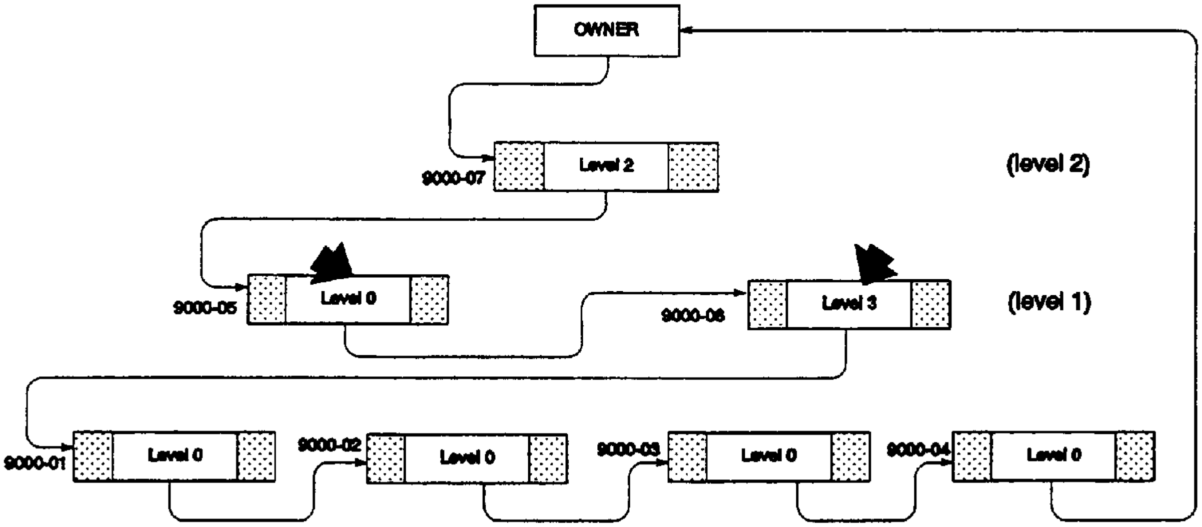


Figure 2.22: IIX Entry with Incorrect Level Numbers

4. Symbolic key errors detected in integrated index records in set

In a set sorted by symbolic key, an SR8 record points to a member record, but the symbolic key contained in the SR8 record does not match the symbolic key in the member record. In Figure 2.23 the db-key in the SR8 record points to the JONES member record, but the symbolic key in the SR8 record is WHITE.

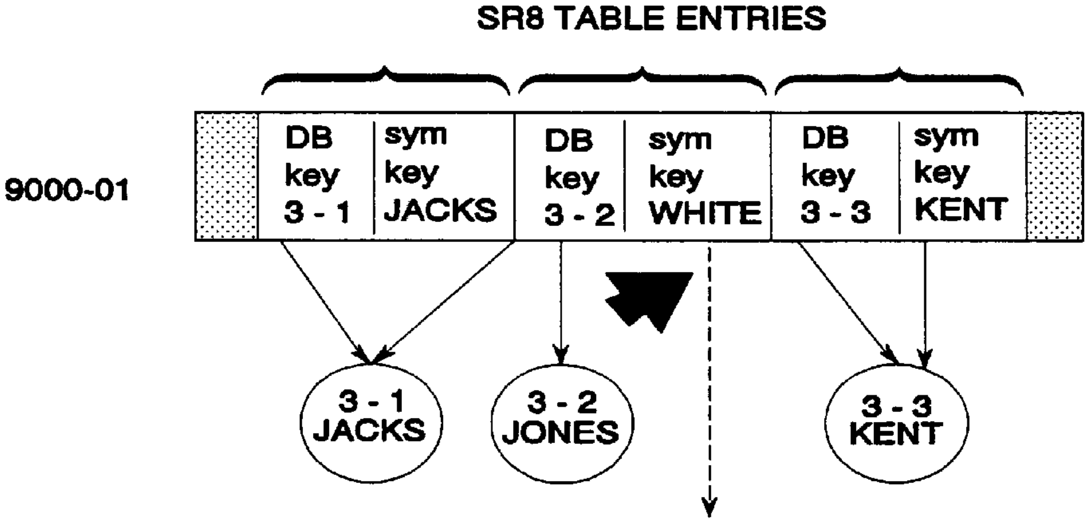


Figure 2.23: Incorrect Bottom Level Symbolic Key

5. Sort key sequence errors detected in integrated index records contained in set

Integrated index sets can be sorted either by db-key or by symbolic key, in ascending or descending order.

In a set sorted by db-key, the last db-key in an SR8 record should match the sort key in the upper level SR8 record just above. For example, in Figure 2.24, in the SR8 record on the right in the bottom level, the last db-key 3-6 does not match the sort key 4-8 in the SR8 record above it.

In a set sorted by symbolic key, the last symbolic key in an SR8 record should match the upper level symbolic key associated with the db-key. For example, in Figure 2.25, the symbolic key CLIFF does not match the symbolic key JONES in the top level SR8 record above.

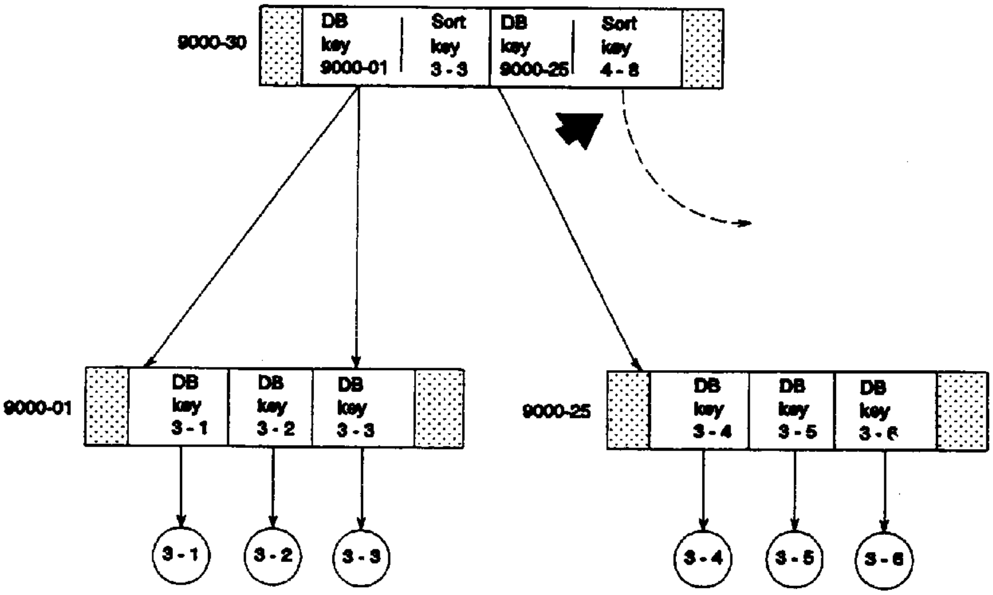


Figure 2.24: Incorrect Upper Level Db-key

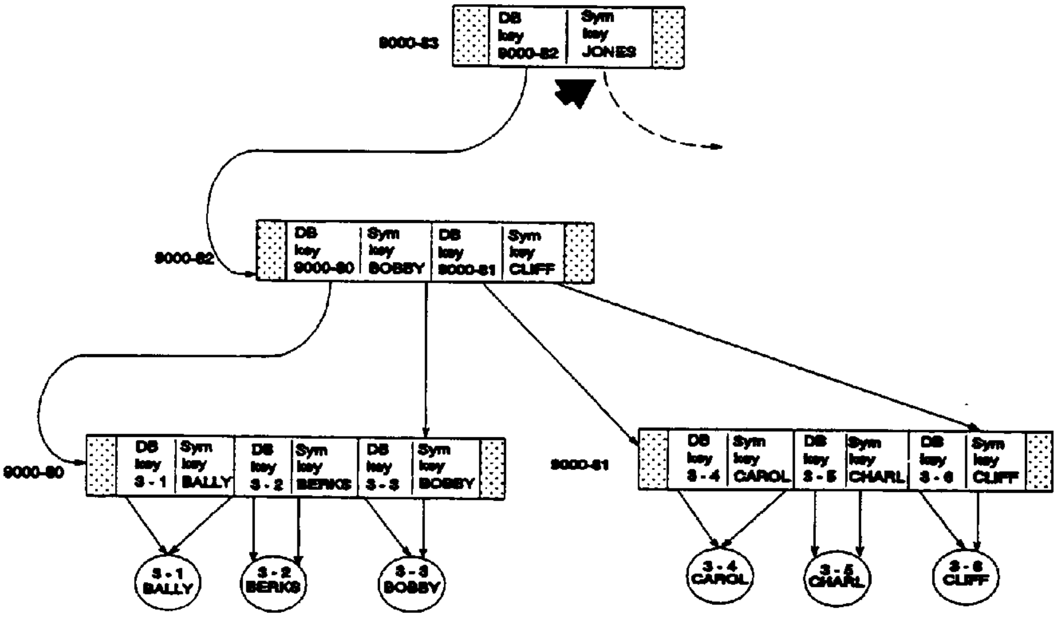
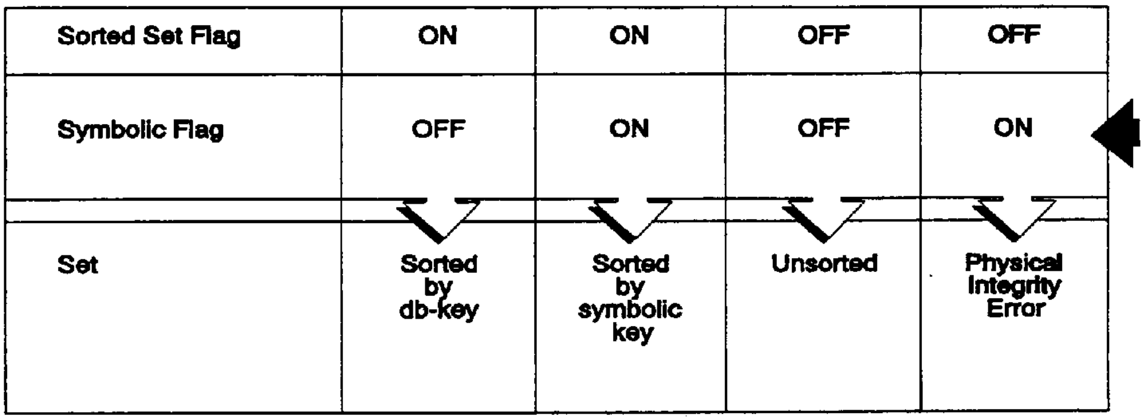


Figure 2.25: Incorrect Upper Level Symbolic Key

6. Sort flag inconsistent with symbolic sort flag in integrated index record

An SR8 record has flags that indicate whether the set is sorted and whether it is sorted by symbolic key. The sorted set flag ON and the symbolic key flag OFF indicates a set sorted by db-key. The sorted set flag ON and the symbolic key flag also ON indicates a set sorted by symbolic key. If the sorted set flag is OFF, the symbolic sort key flag should be OFF. Figure 2.26 shows the combinations of flag settings with their meanings.



Sorted Set Flag	ON	ON	OFF	OFF
Symbolic Flag	OFF	ON	OFF	ON
Set	Sorted by db-key	Sorted by symbolic key	Unsorted	Physical Integrity Error

Figure 2.26: Integrated Index Sorted Flag Settings

7. Integrated index records disconnected from set

A member record is disconnected from an integrated index set. If INDEX is specified in the AUDIT statement, CA-IDMS/DB Audit sweeps the area and creates a table of the db-keys of set member records. During index auditing, CA-IDMS/DB Audit lists all the db-keys of member records of each mandatory-automatic integrated index set. Each member record should have a pair of entries--one obtained during sweep processing, and one obtained during the index auditing. If there is no pair, CA-IDMS/DB Audit reports the member as disconnected.

In Figure 2.27, for example, three of the records in the SWEEP portion of the table are not paired with INDEX records, and two of the records in the INDEX portion of the table are not paired with SWEEP records. The five records are reported as disconnected.

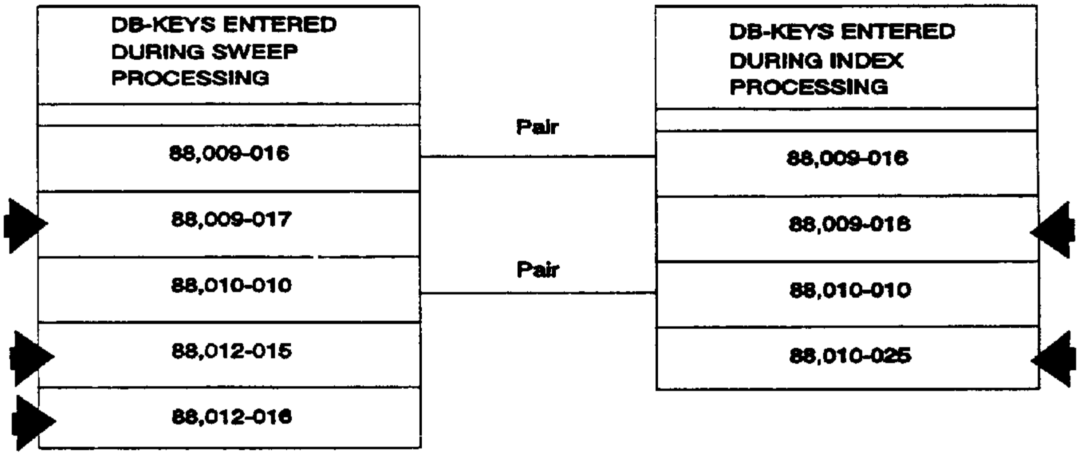


Figure 2.27: Integrated Index Disconnected Records

2.3.4 Orphan Record Errors in Non-Integrated Index Sets

A record designated as an orphan record points at another set or record, but neither the owner of the set nor any member record in the set points back at the orphan record. This type of integrity error is detected during walknext and QuickCheck processing. Do not confuse an orphan record with an integrated index orphan (described earlier). An orphan record **is** an integrity error.

There are two types of orphan record errors:

1. Orphan string points into valid set

CA-IDMS/DB Audit found a record that can not be retrieved when the set is walked from the owner, even though the record points into the set.

For example, in Figure 2.28, records Carol Smith and Ron Barker are connected into the set occurrence owned by the ENGLISH record. When the set is walked from the owner record (ENGLISH), however, only Harry Dean, Jane Doe, and Robin Hayes are retrieved. The Carol Smith and Ron Barker records can only be retrieved if stored CALC, or if the area is swept. CA-IDMS/DB Audit corrects this situation by making Harry Dean point to Ron Barker.

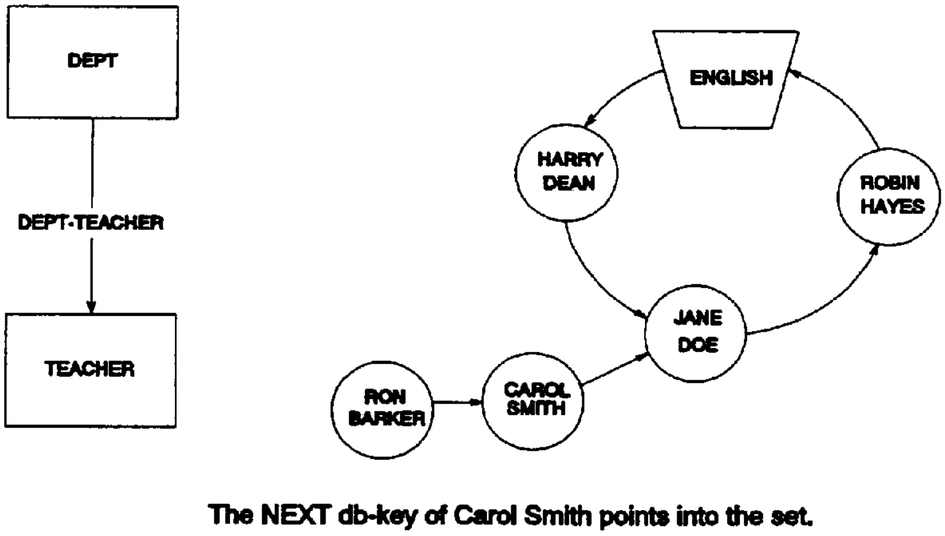
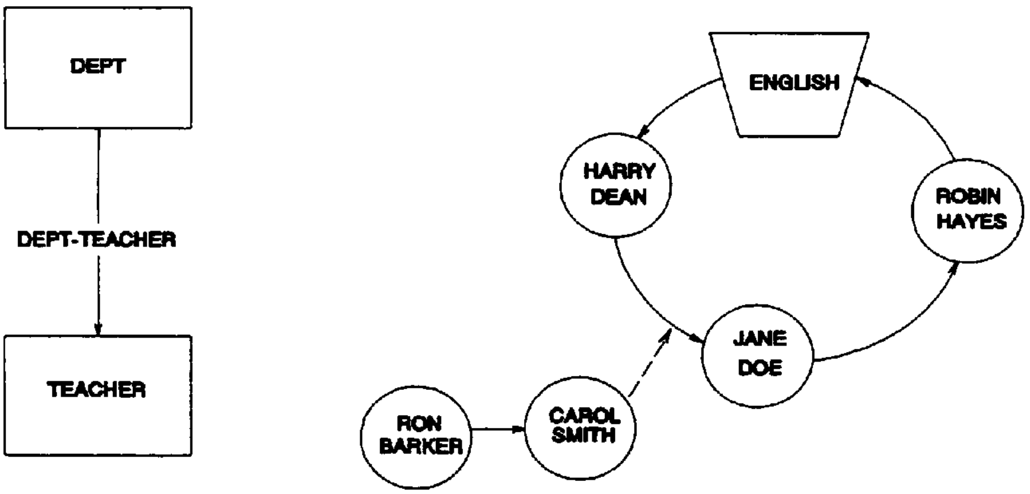


Figure 2.28: Orphan String Points into Valid Set

2. Orphan string is fully disconnected

CA-IDMS/DB Audit could not retrieve an owner record in a set occurrence that was entered either by retrieving one of the member records in an area sweep or by retrieving a member record that was stored CALC. In addition, the member record chain cannot be retrieved by walking from the owner of the set.

For example, in Figure 2.29, TEACHER records Ron Barker and Carol Smith are connected together; however, no valid owner of a DEPT-TEACHER set points to the string. Also, Carol Smith's next db-key is invalid. CA-IDMS/DB Audit cannot correct this situation.



The NEXT db-key of Ron Barker points to Carol Smith, but neither record points into a set, and records in the set do not point to them

Figure 2.29: Orphan String is Fully Disconnected

2.3.5 Fragmented Record Errors

CA-IDMS/DB Audit finds any instance of a fragmented (variable length) record whose fragments could not be properly assembled. For example, in Figure 2.30, the subject record Biology is variable length and should consist of the root portion and one fragment. The pointer within the root portion of the record, however, points to an SR4 record (fragment) that does not exist. CA-IDMS/DB Audit cannot fix errors of this type.

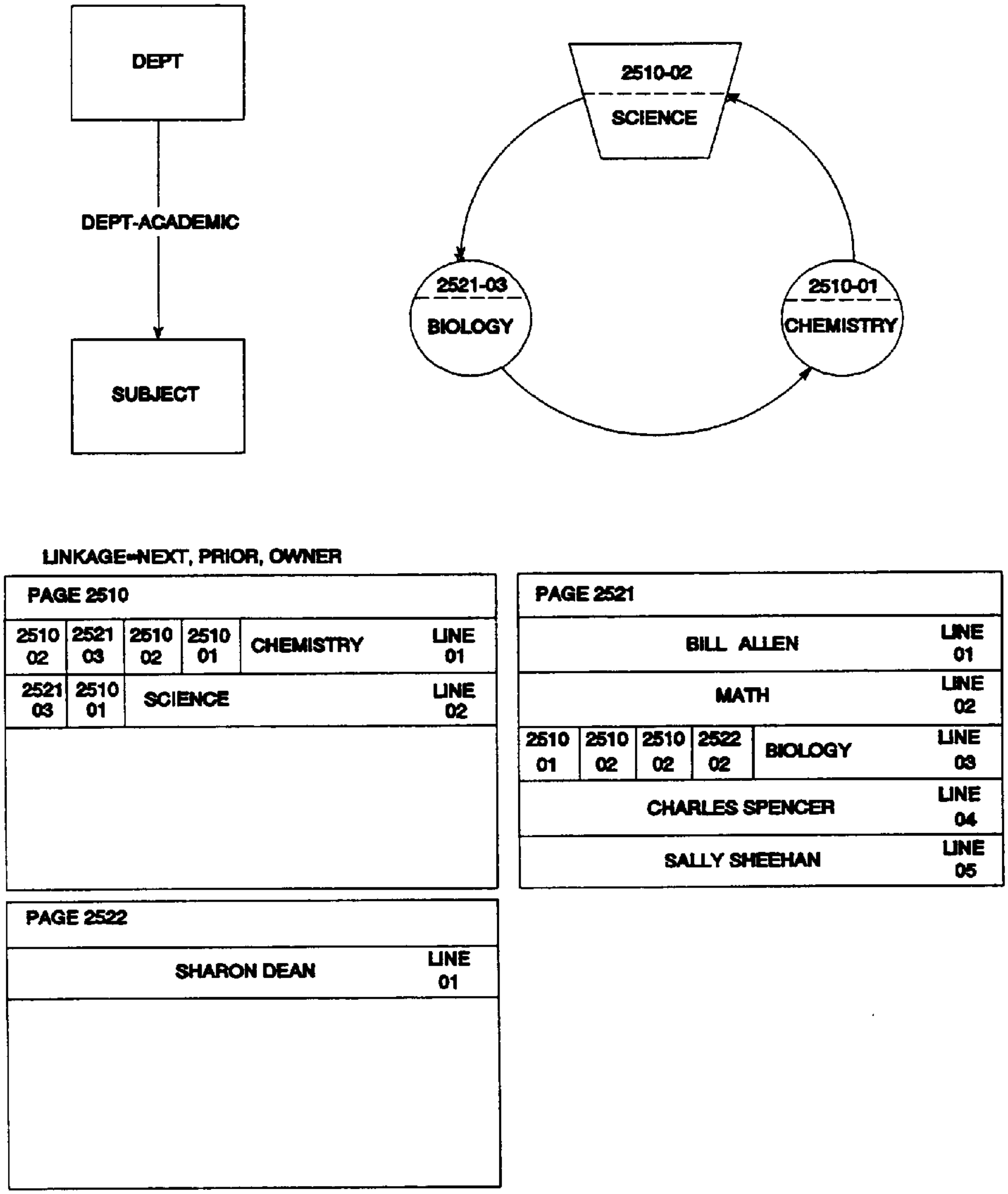


Figure 2.30: Fragmented Record Error

2.3.6 Compressed Record Errors

If there are errors in the compressed portion of a record, it cannot be successfully decompressed. Improper decompression can cause a fragmented record to be improperly assembled. CA-IDMS/DB Audit cannot fix errors of this type.

2.3.7 Page Errors

CA-IDMS/DB Audit detects one type of page error:

Invalid page number

When an area is being swept, CA-IDMS/DB Audit keeps track of the Page-Number of the page that is to be retrieved. A check of the page number within the page header and page footer is performed. For example, in Figure 2.31, the DEPT-REGION low and high page numbers are 5001 and 5100, respectively. When the 12th page is retrieved within the area, the page number of the returned page should be 5012; however, the page footer for page 5012 shows number 5094. CA-IDMS/DB Audit corrects the page footer to be 5012.

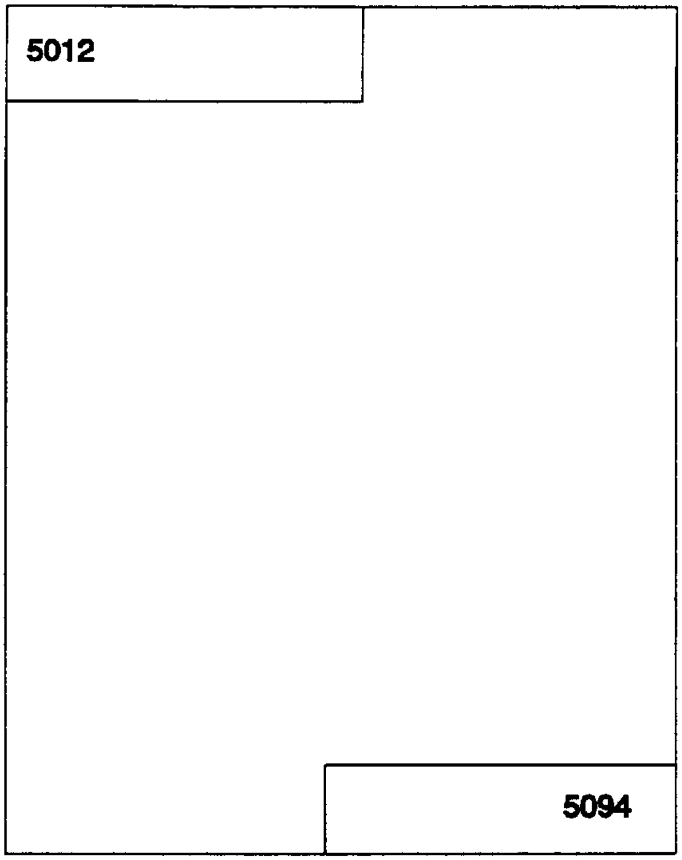


Figure 2.31: Invalid Page Number

2.4 Auditing Methods

CA-IDMS/DB Audit has two methods of auditing:

- Standard audit
- QuickCheck audit

Standard auditing examines your application databases and data dictionary for all types of physical integrity errors. Although Standard auditing is the most comprehensive, it requires numerous I/Os and potentially long processing times.

QuickCheck is a much faster processing alternative for detecting common physical integrity errors. QuickCheck requires significantly fewer I/Os and less CPU time because it does not walk every set occurrence to detect errors. Instead, during area sweeps QuickCheck creates pointer extract records that are then sorted. If CA-IDMS/DB Audit detects any mismatched pointer extract records, it then walks only those sets that are in error. Finally, errors found by QuickCheck are reported and most can be corrected using the same fix parameters specified in Standard auditing.

2.4.1 QuickCheck vs. Standard Audit

Deciding whether to use QuickCheck or Standard auditing depends upon the nature of the audit. Use QuickCheck when time is limited and your main concern is locating next, prior, or owner (N/P/O) pointer errors. Because Standard auditing is more thorough, it is the preferred alternative when you need to search for all error types.

Figure 2.32 summarizes the major differences between QuickCheck and Standard auditing. Features that are supported by one method or the other are marked with an X. Several features are supported under QuickCheck, but only under certain circumstances. These features are qualified by a brief explanation.

Features	Standard	Quickcheck
Detects N/P/O errors in user sets	X	X - faster performance
Detects N/P/O errors in IIX sets	X	X - faster performance
Detects ownerless sets	X	if located in N/P/O error set
Detects sets with 2 or more owners	X	if located in N/P/O error set
Detects sorted set sequence errors	X	if located in N/P/O error set
Detects fully disconnected records	with WALKNEXT and DISCONNECT	X - faster performance

Detects incorrect CALC membership	X	
Detects logically deleted records	X	if located in N/P/O error set
Detects non-N/P/O IIX set errors	X	
Detects orphan records	with WALKNEXT	X - much faster performance
Walks all sets	X	
Walks error sets only		X
Supports partial page range audits	X	if neither USERSETS nor CALCSETS is specified
Supports fast scan buffering	X	X
Supports pre-audit scanning	X	X
Supports inclusive processing	X	X

Figure 2.32: Summary of Standard and QuickCheck Features

2.5 Audit Phase

In the audit phase, CA-IDMS/DB Audit searches through a database looking for physical integrity errors. It creates an extract file containing each physical integrity error found and each logically deleted (LDEL) record found. This extract file is used by the fix and reports phases for further processing.

CA-IDMS/DB Audit uses several audit processes:

- Sweep
- Set Walk
- Walknext
- Set Disconnection
- Integrated Index
- Page Audit

The user initiates one or more of these processes through parameter statements and options. Although Chapter 3, “Parameters” all parameters in detail, this chapter describes the audit processes in general and provides diagramed examples.

2.5.1 Sweep Processing

The primary method used by CA-IDMS/DB Audit to access a user's database is sweep processing. Using sweep processing, CA-IDMS/DB Audit starts retrieving the records in an area defined by the subschema. It retrieves all records according to AREA, SET, and RECORD parameter statements. After a record is retrieved, CA-IDMS/DB Audit refers to the parameters you specify to determine how to audit the record and its sets.

Page headers and footers are also audited during sweep processing if you specify the PAGES option in the AUDIT statement. When CA-IDMS/DB Audit retrieves a new page during sweep processing, it examines the header and footer before looking at the records on the page.

During sweep processing, CA-IDMS/DB Audit retrieves records within an area by db-key, in numerical order. For example, in Figure 2.33, assume that the DEPT-REGION area, DEPT and TEACHER record types, and DEPT-TEACHER sets have been copied into the subschema. When sweeping the DEPT-REGION area, the records HISTORY, Harry Dean, ENGLISH, Carol Smith, and John Doe will be retrieved in that order.

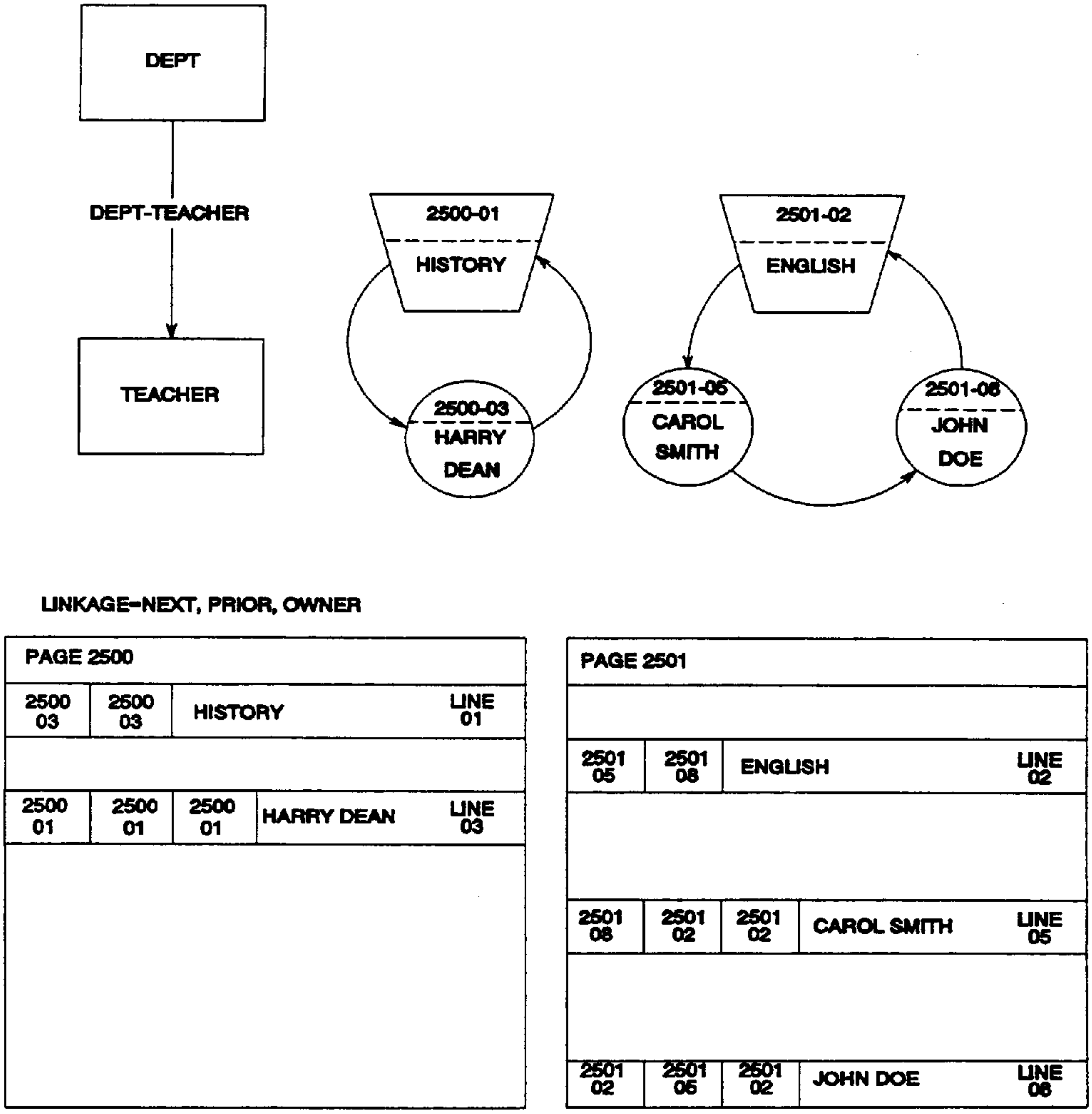


Figure 2.33: Sweep Processing

2.5.2 Set Walk Processing

During Standard auditing, once a record has been selected by the sweep process, CA-IDMS/DB Audit can perform set walk processing. The main function of set walk processing is to ensure the integrity of a set occurrence. If the record retrieved by CA-IDMS/DB Audit is the owner of one or more set occurrences, CA-IDMS/DB Audit walks the set from the owner record until it reaches the end-of-set.

If CA-IDMS/DB Audit encounters any physical integrity errors during set walking, it returns to the owner record and walks the set again, creating an extract record for each record found in the set. The file of extract records is saved for later use by the FIX phase or the REPORTS phase.

The two options used to initiate set walking are USERSETS and CALCSETS. The SETLIMIT and DBKEYTBL options provide additional controls on set walking.

2.5.2.1 USERSETS Option

The USERSETS option initiates set walking for user-defined sets and integrated index sets.

During USERSETS processing, CA-IDMS/DB Audit walks each set only one time if it does not detect any next, prior, or owner pointer errors. If a record is retrieved that does not own any set occurrences, it is bypassed.

In Figure 2.34, only records that are owners of sets initiate a set walk during the sweep of the DEPT-REGION area. Carol Smith and John Doe do not initiate a set walk by CA-IDMS/DB Audit, because they are not owners of any sets. When the owner record ENGLISH is retrieved, the DEPT-TEACHER set is walked (that is, ENGLISH, Carol Smith, John Doe, and finally ENGLISH again). In addition, the DEPT-ACADEMIC set is walked (that is, ENGLISH, American Literature, English Literature, Remedial English, and ENGLISH).

During QuickCheck auditing, USERSETS also looks for orphans. During Standard auditing, orphans are detected through WALKNEXT processing (discussed in detail later in 2.5.3, “Walknext Processing.”)

2.5.2.2 CALCSETS Option

If you specify CALCSETS in the AUDIT statement, each CALC set is walked to audit for next and prior pointer errors, in a manner similar to walking all other sets. Also, during STANDARD auditing, each CALC record is checked to ensure that it is a member of the correct CALC set.

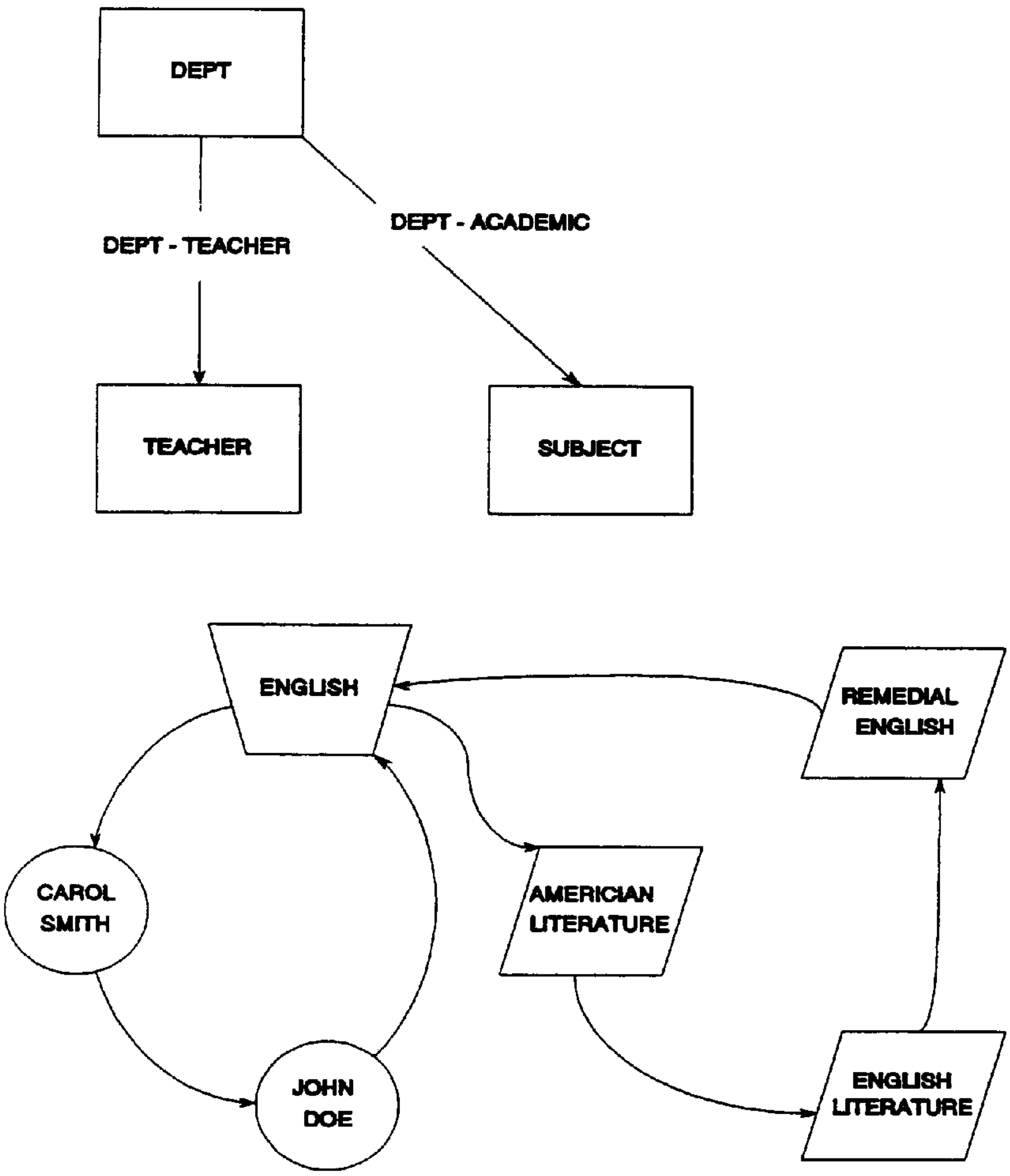


Figure 2.34: Set Walk Processing

2.5.2.3 SETLIMIT Option

To prevent CA-IDMS/DB Audit from entering into a set loop while set walking, you can specify a limit on the number of records (in the set) that CA-IDMS/DB Audit retrieves before assuming that a set loop condition exists. For example, in Figure 2.35, assume that SETLIMIT=5. When CA-IDMS/DB Audit walks the DEPT-ACADEMIC set owned by ENGLISH, the following records are retrieved: ENGLISH, American Literature, English Literature, Remedial English, American Literature, English Literature, and Remedial English. CA-IDMS/DB Audit terminates processing of the set occurrence at this point because six member records have been retrieved, one more than the SETLIMIT value.

Note: CA-IDMS/DB Audit does not fix any sets during the fix phase if the SETLIMIT value is reached for any one set occurrence. Specifying a SETLIMIT value that is too small can cause CA-IDMS/DB Audit to stop processing a set occurrence prematurely and thus prevent fixing of any sets.

2.5.2.4 DBKEYTBL Option

To detect a set loop and allow fixing of these errors, you can specify DBKEYTBL in the AUDIT statement. CA-IDMS/DB Audit maintains a table of all member record db-keys retrieved for each set occurrence during set walk processing. Whenever a record's next db-key is found in the table already, CA-IDMS/DB Audit stops processing the set. In Figure 2.35, when auditing the DEPT-ACADEMIC set, the following member records are retrieved and their db-keys placed in the table: American Literature, English Literature, and Remedial English. Since Remedial English's next db-key is already in the table, CA-IDMS/DB Audit recognizes that the set contains a loop and terminates the set-walk for that set occurrence.

Note: Even if DBKEYTBL is not specified in the AUDIT statement, CA-IDMS/DB Audit activates the DBKEYTBL function during the creation of extract records to determine whether or not a next or prior db-key points to a record previously retrieved in the set. Therefore, you must plan storage for DBKEYTBL.

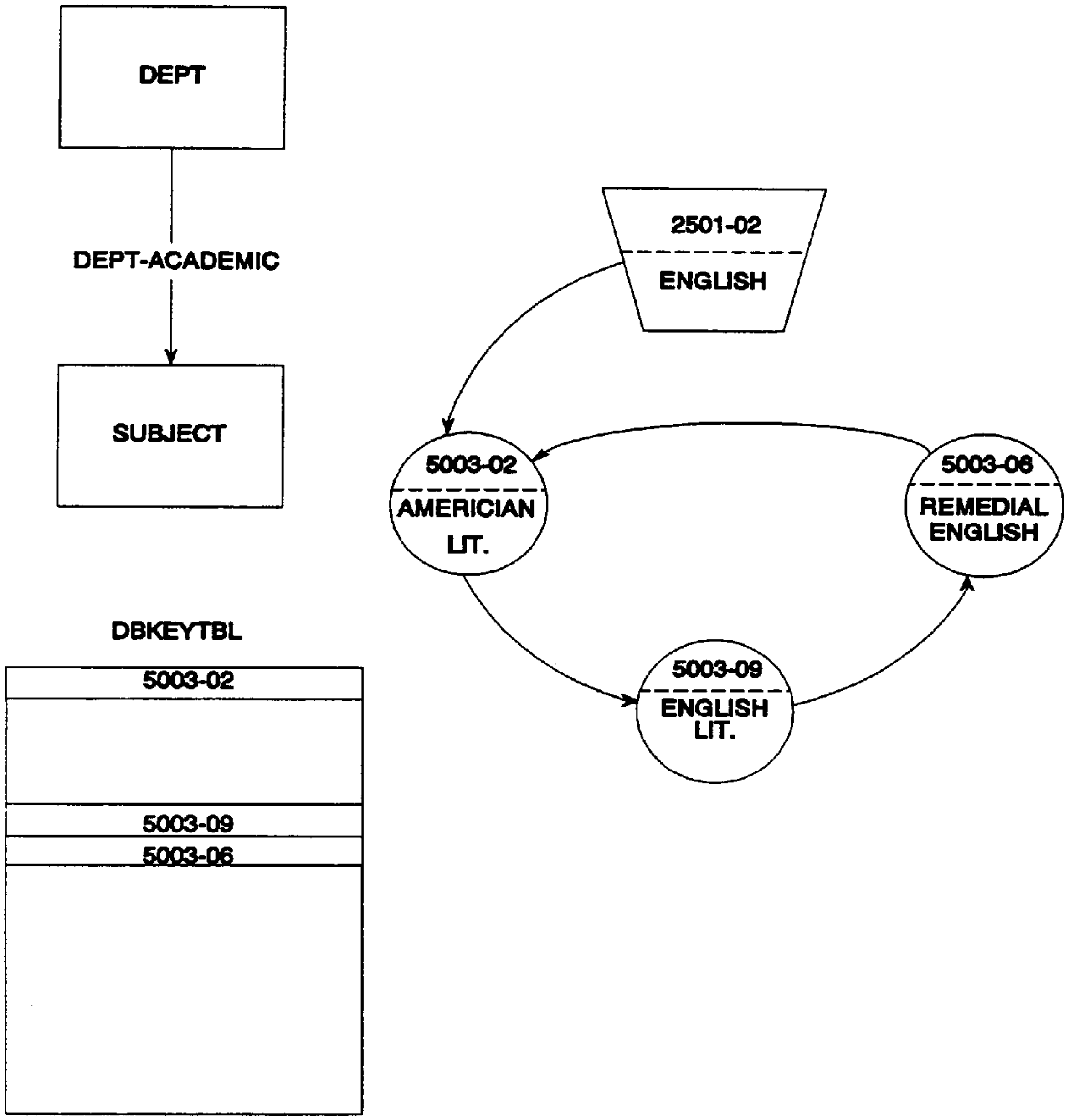


Figure 2.35: Set Loop

2.5.2.5 Prior Pointers

When walking a set containing prior pointers, if the prior db-key of a record occurrence does not point back to its prior record in the set, CA-IDMS/DB Audit saves the db-key of the current record in a save table and continues set walking. At end-of-set, CA-IDMS/DB Audit retrieves each saved db-key and attempts to walk prior from that point. This process of walking, saving, and retrieving db-keys is continued until all saved db-keys have been processed.

For example, in Figure 2.36, CA-IDMS/DB Audit walks a set occurrence owned by ENGLISH. No error is detected until the David Lee record is retrieved: the prior db-key does not point to John Doe.

Since the David Lee record is the last member of the set and an integrity error was detected in the set, CA-IDMS/DB Audit returns to the owner record, ENGLISH, and walks the set again, creating extract records for each record occurrence retrieved. When the David Lee record is retrieved, the record db-key is placed in the save table. CA-IDMS/DB Audit continues set walking until end-of-set is reached. CA-IDMS/DB Audit then retrieves the saved db-key and begins walking prior from the saved db-key to its prior record, Joan Brown.

The following situations demonstrate the effect of specifying DBKEYTBL or not:

- When DBKEYTBL is specified: because Joan Brown's prior db-key (2501-08) is already in the DBKEYTBL, CA-IDMS/DB Audit stops walking prior at this point
- When DBKEYTBL is not specified: CA-IDMS/DB Audit continues walking prior to the records John Doe and Carol Smith until it reaches the owner record ENGLISH, or until the SETLIMIT value is reached

In either case, CA-IDMS/DB Audit then checks the save table for additional saved db-keys. After all saved db-keys have been processed, CA-IDMS/DB Audit stops creating extract records for the set.

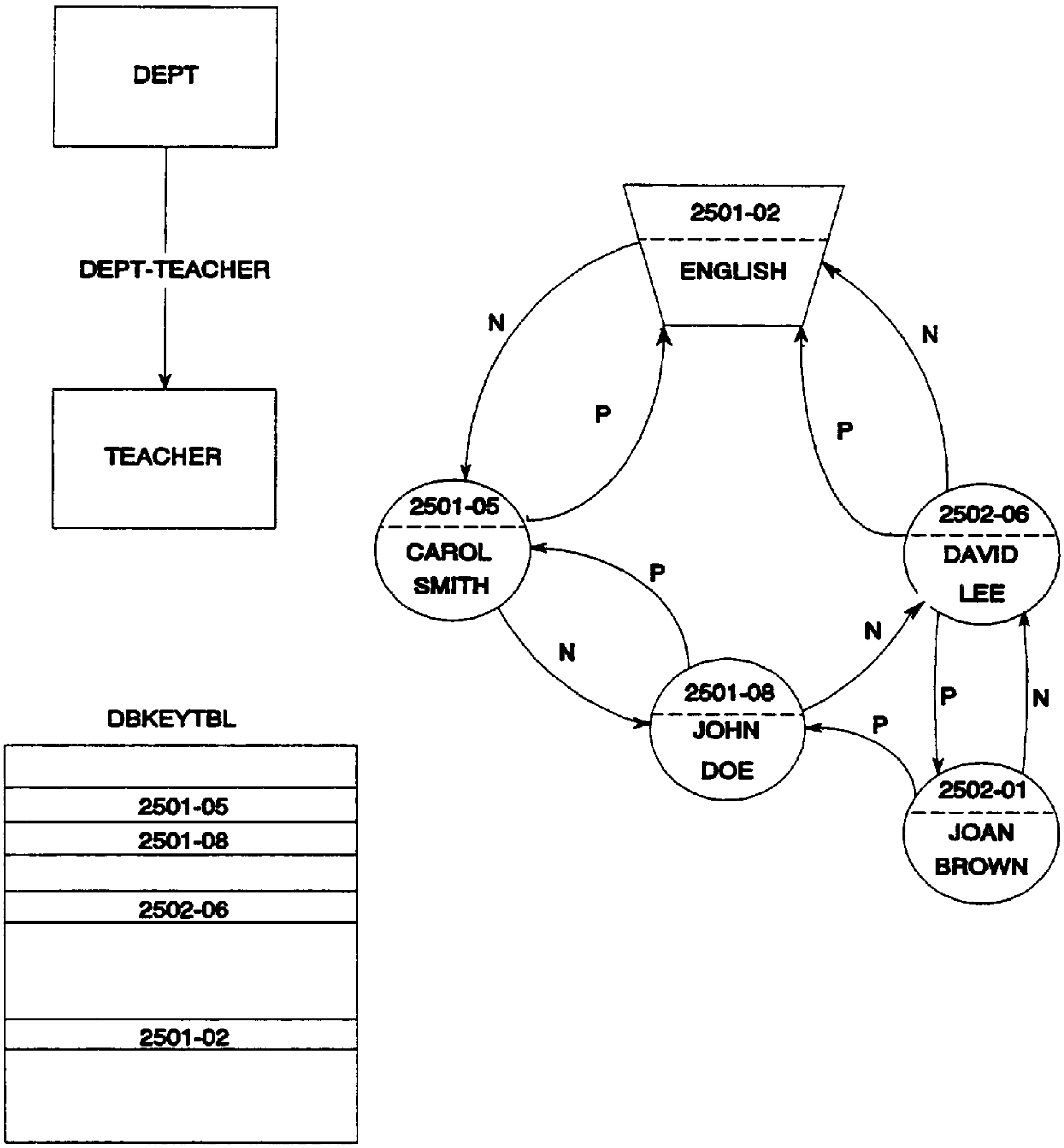


Figure 2.36: Set Walk, Prior Pointers

2.5.3 Walknext Processing

An orphan record is one type of error that cannot be detected and adequately diagnosed by simply walking each set once from its owner. If a record points into a set of which it is a member, but neither the owner nor any of the members point to the record, then it is referred to as an orphan record.

One way to detect orphan records is to walk each set, not only when the owner record is detected during sweep processing, but also when each member record of the set is retrieved. In other words, as each record is retrieved during sweep processing, not only are all set occurrences owned by the record walked, but all sets where the current record is a member are walked.

For Standard auditing, you initiate walknext processing by specifying WALKNEXT in the AUDIT statement. For QuickCheck auditing, orphan detection occurs when you specify the USERSETS option. You cannot specify WALKNEXT with QuickCheck auditing.

If you initiate walknext processing, CA-IDMS/DB Audit walks all sets of which the retrieved record is a member. If an owner record is encountered, CA-IDMS/DB Audit begins walking again from the owner record until it reaches either the original retrieved record or the owner record for a second time. For example, in Figure 2.37, when record GR 79-1 is retrieved during the area sweep, CA-IDMS/DB Audit walks forward until the owner record (Bill Smith) is retrieved. Once the owner record is detected, CA-IDMS/DB Audit walks forward again until record GR 79-1 is retrieved for a second time.

In order to understand the impact of using WALKNEXT, assume that there are 6,000 STUDENT records and 24,000 GREPORT records. If the STUDENT-GREPORT set is audited using USERSETS, CA-IDMS/DB Audit will access 30,000 records (6000 owners plus 24000 members; average set length is 4). When auditing with WALKNEXT, CA-IDMS/DB Audit will access 150,000 records (30,000 set walking from the owners and 120,000 set walking from each member record). If GREPORT members are not stored VIA the STUDENT-GREPORT set, CA-IDMS/DB Audit will generate 150,000 I/Os, based on an average of 4 member records per set times 30,000 record occurrences, plus 30,000 I/Os for the set walk process.

Note: If you specify WALKNEXT, each set is walked as many times as there are member records. This multiple walking causes a substantial increase in processing time. You can, however, search for orphans through QuickCheck with the USERSETS option and use significantly less processing time.

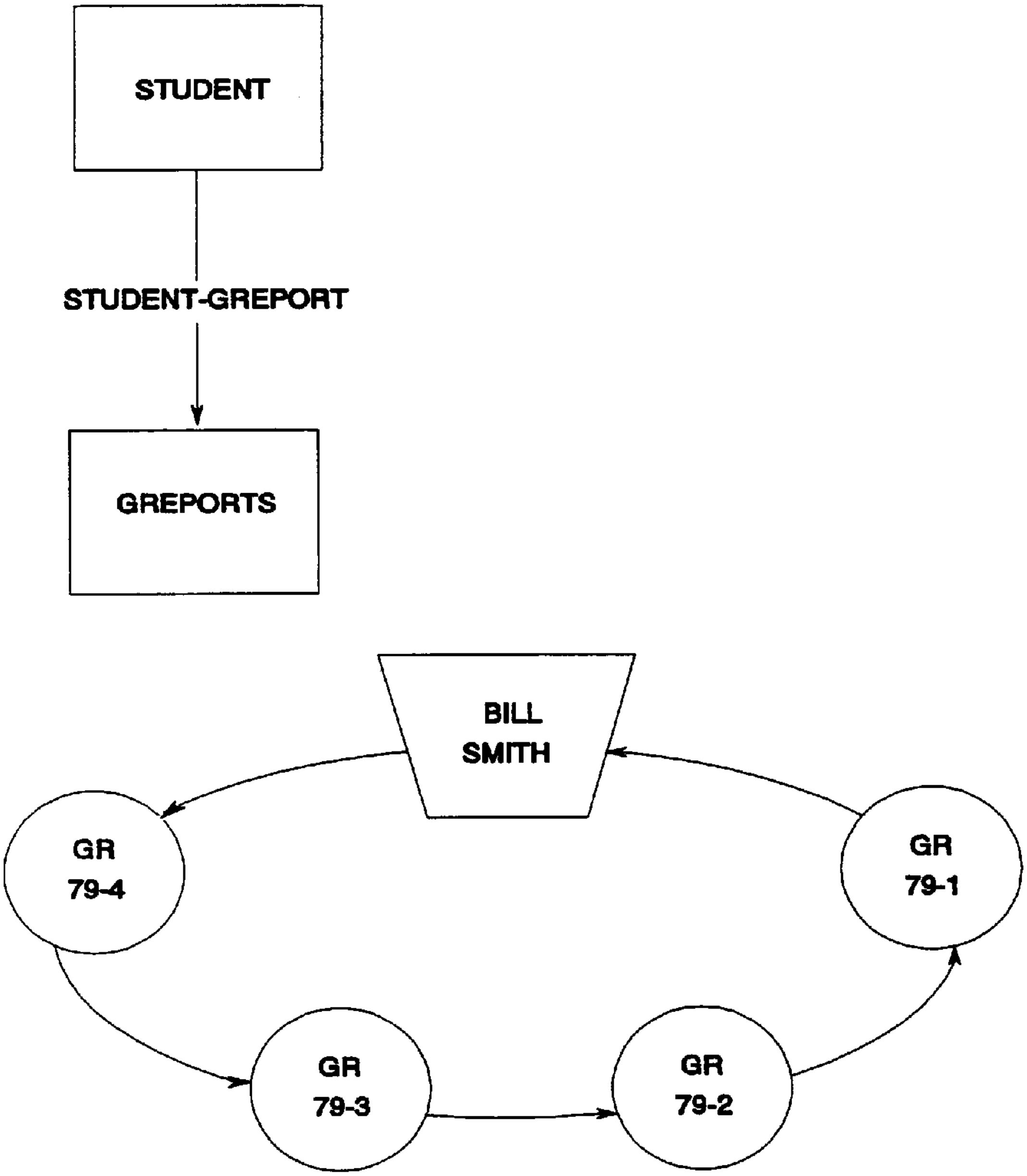


Figure 2.37: Walknext Processing

If an owner record is obtained for the second time, the retrieved record is an orphan. For example, in Figure 2.38, when the Creative Writing record occurrence is retrieved, CA-IDMS/DB Audit detects an orphan record. CA-IDMS/DB Audit walks from Creative Writing to Remedial English back to ENGLISH. Then it walks from ENGLISH to American Literature to English Literature to Remedial English back to ENGLISH. This sequence indicates that Creative Writing is connected into the set owned by ENGLISH, but cannot be retrieved by walking from the ENGLISH record. Therefore, Creative Writing is an orphan record.

2.5.3.1 NOSWEEP Option

If the owner of the set is in a NOSWEEP area, the set will not be walked. In that case, CA-IDMS/DB Audit cannot determine whether records in another area are orphan records of sets whose owners are in the NOSWEEP area.

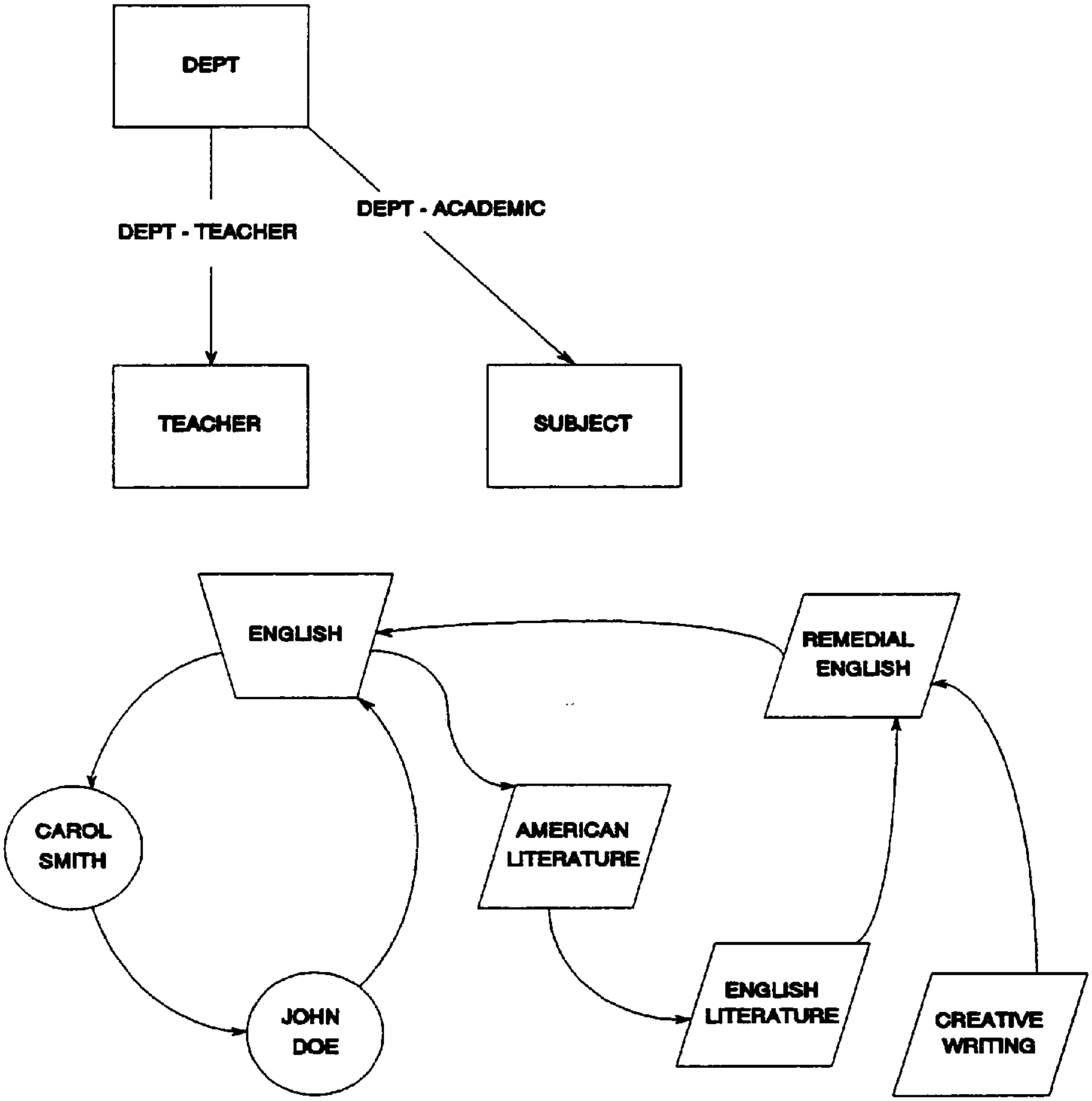


Figure 2.38: Orphan Record (Partially Disconnected)

2.5.4 Set Disconnection Processing

During sweep processing, when each record is retrieved, CA-IDMS/DB Audit checks to see if the record is a member of a set to be audited for disconnections, or if the record type is to be audited for disconnections. A record is disconnected if no records that are members of a set point to it and it does not point into the set.

No search for disconnections is made if you specify DISCONNECT=NONE in the AUDIT statement.

2.5.4.1 Verifying Pointers

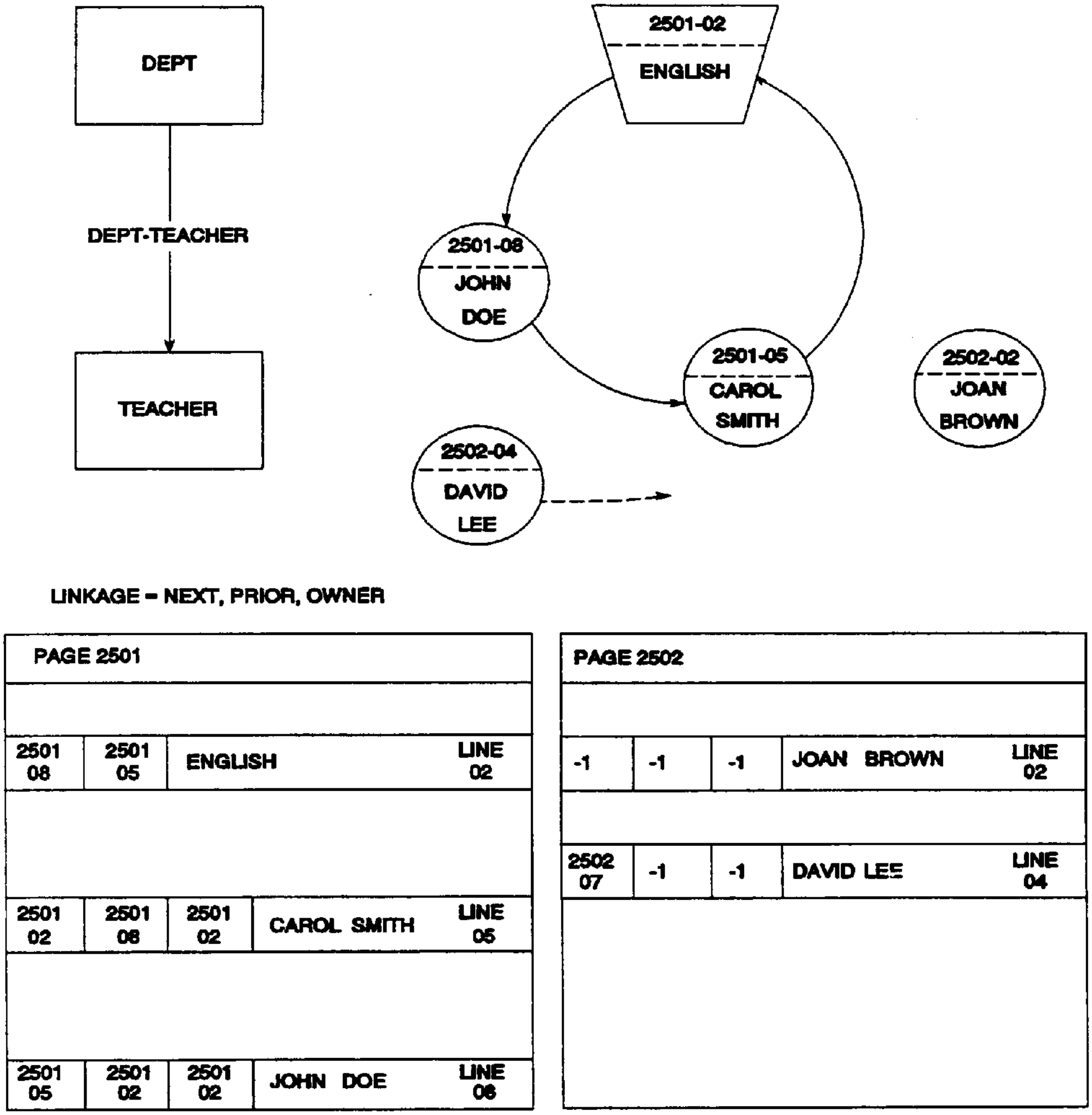
When a record is to be audited for set disconnections, each pointer is verified by finding the indicated record. For example, if there are 24,000 GREPORT records with next, prior, and owner pointers in the STUDENT-GREPORT set, then CA-IDMS/DB Audit verifies each pointer (72,000 pointers) by finding 72,000 records.

Two disconnected records are illustrated in Figure 2.39. If all of the pointers of a record are null, the record is fully disconnected. If all of the pointers of a record are a combination of invalid and null pointers, the record is partially disconnected. Because all of the pointers of the Joan Brown record are null (indicated by -1), it is fully disconnected. The David Lee record is partially disconnected, because its only pointer that is not null does not point into a set.

Note: Because of the large number of I/Os and calls to DBIO, DISCONNECT can increase processing time significantly. You can, however, reduce processing time through QuickCheck. QuickCheck DISCONNECT looks for fully disconnected records only. QuickCheck USERSETS looks for partially disconnected records only.

2.5.4.2 DISCONNECT with WALKNEXT

WALKNEXT takes precedence over DISCONNECT when both are specified in the same run. If an error is found in both walknext processing and in set disconnection processing, the record in error will be reported as an orphan record. As in Figure 2.39, David Lee would be reported as an orphan record if you specified WALKNEXT, as well as Disconnect.



PAGE 2501				
2501 08	2501 05	ENGLISH		LINE 02
2501 02	2501 08	2501 02	CAROL SMITH	LINE 05
2501 05	2501 02	2501 02	JOHN DOE	LINE 08

PAGE 2502				
-1	-1	-1	JOAN BROWN	LINE 02
2502 07	-1	-1	DAVID LEE	LINE 04

Figure 2.39: Orphan Record (Fully Disconnected)

2.5.5 Integrated Index Processing

If INDEX is specified in the AUDIT statement, CA-IDMS/DB Audit verifies each integrated index entry against its database record occurrence. You cannot specify the INDEX option with QuickCheck auditing.

To detect integrated index orphans, CA-IDMS/DB Audit examines the upper level pointers in each SR8 record.

CA-IDMS/DB Audit also verifies the entries in the SR8 records of the integrated index set by trying to retrieve any records pointed to by the db-keys in the entries.

CA-IDMS/DB Audit verifies:

- Upper level pointers
- Db-keys:
  - Record not found
  - Record type incorrect
  - Record not within page range
  - Record located in non-readied area
- Symbolic keys
- Sort key sequences (set sorted by db-key or symbolic key)
- Level numbers
- Sort flags

A check for disconnections of member records is made when CA-IDMS/DB Audit sweeps the area looking for mandatory-automatic member records of integrated index sets. It then compares the list of the member records with a list of member records found while walking down from the owners of integrated index sets.

2.5.6 Page Audit Processing

CA-IDMS/DB Audit performs page auditing when you specify PAGES in the AUDIT statement. Whenever a new page is retrieved during sweep processing, CA-IDMS/DB Audit checks the page-number within the page-header and page-footer against the requested page number.

Note: If a test SUBSCHEMA/DMCL is used to audit a production database (or vice versa), a large number of pages can be incorrectly flagged as being in error. Make sure that the correct SUBSCHEMA/DMCL and load/core image library are specified when requesting PAGES.

2.6 Specifying Database Auditing

CA-IDMS/DB Audit determines which parts of the database to audit by comparing the subschema and the parameters that you specify. The subschema must define these entities:

- Areas to be swept
- Areas containing owners of sets to be audited
- Areas containing members of sets to be audited
- Set types to be audited
- Owner record types of sets to be audited
- All member record types of sets to be audited
- Index sets to be audited

2.6.1 Area Sweeping

If there are no restrictions by AREA, SET, or RECORD statements, an area is swept if any of the following AUDIT parameters are specified:

```
USERSETS, if owners of sets are in the area
WALKNEXT
PAGES
DISCONNECT
CALCSETS
INDEX
```

An area is not swept if only USERSETS is specified and no owners of sets to be audited are in the area. Access is still available, however, to member records of audited sets whose owners are in other areas.

2.6.2 Set Auditing

When you specify USERSETS, user-defined sets named in the subschema are audited if the set types, their owner record types, all member record types, and areas containing all of those record types are defined in the subschema.

2.6.3 Limiting the Audit

You can include or exclude parts of the database for auditing by using the STARTWITH option in conjunction with AREA, SET, and RECORD statements. Realize, however, that a limit placed on an area, record, or set may unintentionally affect the auditing of another set. For example, if an AREA is designated NOSWEEP, walknext processing is not performed on any sets whose owners are in the NOSWEEP area.

2.7 Parameter Controlled Functions

By specifying various parameters, you can direct CA-IDMS/DB Audit to perform the following functions:

- Report disconnections. Records disconnected from sets are detected and reported when you specify either the ALL or the EXPLICIT option of the DISCONNECT parameter.
- Stop all processing after detecting a specified number of errors by using the STOPAFTER option of the AUDIT parameter statement.
- Audit the CA-IDMS CALC set, by specifying CALCSETS.
- Audit user-defined sets and integrated index sets by specifying USERSETS.
- Verify each integrated index entry against its database record by specifying INDEX.
- Restrict the portion of an area to be swept by the audit phase, by using the BEGIN and END options in the AREA parameter statement. By using this facility, you can audit a large area in sections. Fixing will not be allowed, however, unless the owners of all sets involved in integrity errors are in the areas audited.
- Restrict the auditing of areas, records, and sets defined in the subschema by using NOSWEEP, NOACCESS, and NOAUDIT.
  - NOSWEEP prevents CA-IDMS/DB Audit from accessing specific records or all the records stored in an area during CA-IDMS/DB Audit's sweep processing. It does not prevent CA-IDMS/DB Audit from accessing member records as part of USERSETS auditing. If owners of sets are in a NOSWEEP area, however, those sets are not processed during walknext processing.
  - NOACCESS prevents CA-IDMS/DB Audit from accessing a specified area or record under any circumstances. It will not attempt to sweep an area or access a record during sweep processing, nor will it access any records stored in an area, or any sets whose owners or members are in the area, during set walk auditing.
  - NOAUDIT prevents CA-IDMS/DB Audit from performing any of its audits on the specified set.

2.8 Fix Phase

CA-IDMS/DB Audit fixes most physical integrity errors found in the audit phase. You can specify either simulated fixing or actual updating of the database.

2.8.1 Errors Fixed

By using information contained in the extract file, CA-IDMS/DB Audit fixes the following physical integrity errors:

- User-defined sets and CALC sets
  - Null db-keys
  - Db-keys that point outside of the subschema page range
  - Next, prior, or owner db-keys that point incorrectly
  - Sort keys out of sequence
  - Set loops
  - Orphan records not properly connected to a set
  - CALC records not properly connected to the CALC set
- Pages: inconsistencies in header or footer page number
- LDELs: CA-IDMS/DB Audit can erase an LDEL if it is not part of a set occurrence in error or the owner of an empty set
- Integrated index sets
  - Next, prior, and owner db-keys that point incorrectly
  - Upper level pointers in orphan records
  - Orphan count

2.8.2 Limits

If CA-IDMS/DB Audit reaches any STOPAFTER, SETLIMIT or DBKEYTBL limits, it does not fix any detected errors because the audit may not be complete.

2.8.3 Errors Only Detected

There are some types of physical integrity errors that CA-IDMS/DB Audit cannot fix:

- FROM/THRU page range errors
- Duplicate sort keys in a set occurrence where duplicates are not allowed
- Fully disconnected record occurrences, including orphan strings
- Fragmented (variable length) records that could not be properly assembled
- Compressed records that could not be properly decompressed

- Integrated index errors other than orphans, orphan counts, and next, prior, or owner pointers (upper level pointers, db-keys, symbolic keys, sort key sequences, level numbers, or sort flags)

You can fix many of these errors by using the information produced by CA-IDMS/DB Audit in conjunction with the CA-IDMS utility IDMSBCF, FIX PAGE function. See individual message descriptions in Chapter 6, “Messages” for more information.

2.8.4 Logical Integrity

CA-IDMS/DB Audit fixes only physical integrity errors. Logical integrity errors cannot be fixed because CA-IDMS/DB Audit does not know the logical basis used to store the data on the database.

For example, in Figure 2.40, the STUDENT-REPORTS set contains two record types: GR and DR. Each record type is sorted in descending order by year and grading period. DR records are stored whenever necessary and should occur in the set after the GR records.

Assume that one of the next pointers is broken. CA-IDMS/DB Audit fixes the set as shown in Figure 2.41. If logically the DR records are to follow the GR records for the same year and grading period, then the “fixed” set occurrence is not in the correct logical sequence. Nevertheless, the physical integrity of the set is correct. Logical integrity can be fixed by a user program--a correction that may not have been possible before the physical integrity was fixed.

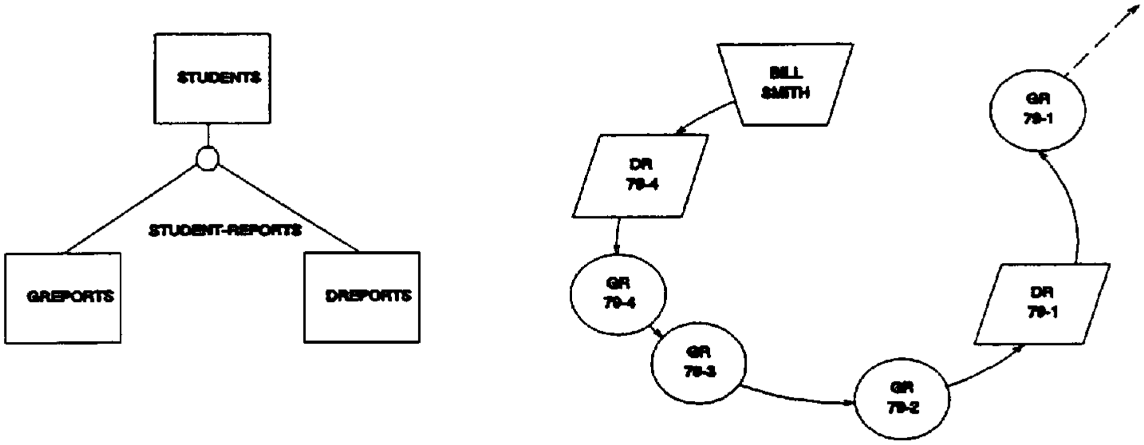


Figure 2.40: Physical Integrity Error

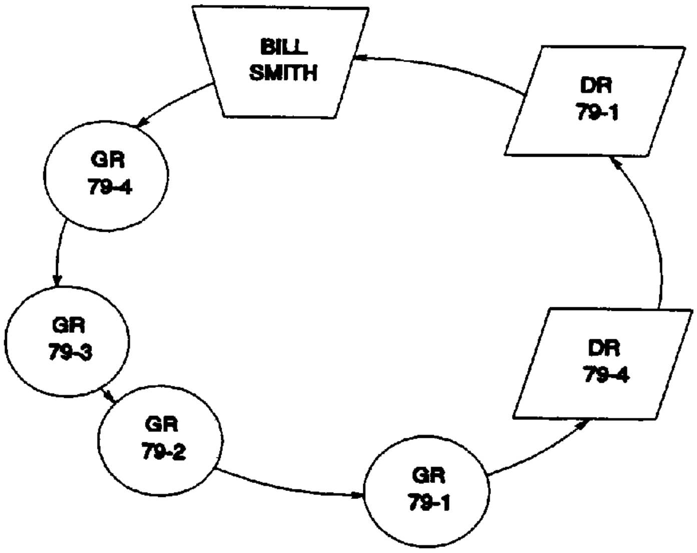


Figure 2.41: Physical Integrity and Logical Integrity

2.8.5 Multiple Owner Fixing

When a record is found in more than one set, CA-IDMS/DB Audit places the record in the set indicated by its owner pointer, or in the set where it appears closest to the owner record. For example, in Figure 2.42, when walking from owner record ENGLISH, member records John Doe, Joan Brown, James Jones, and Paul Cary are retrieved before an error is detected. Walking from owner record MATH, member records David Kraft, James Jones, and Paul Cary are retrieved. Record occurrences James Jones and Paul Cary appear in both sets.

If such a record has an owner pointer that matches the db-key of an owner record, the member record is placed in that owner's set.

If the record does not have a valid owner pointer, the record is placed in the set where it is nearest the owner record. In Figure 2.42, James Jones and Paul Cary are closest to the MATH record (that is, they are the 2nd and 3rd records in the MATH set as opposed to 3rd and 4th records in the ENGLISH set). Figure 2.43 shows how the sets will be fixed by CA-IDMS/DB Audit, by placing James Jones and Paul Cary in the MATH set.

If a record does not have a valid owner pointer and it is equidistant from multiple owner records, it is placed in the set whose owner has the lowest db-key.

The fixed sets are unbroken, but either James Jones or Paul Cary, or both, may belong in the ENGLISH set. You should verify the logical integrity of the sets by examining the After Image Report to see how the error is corrected by CA-IDMS/DB Audit.

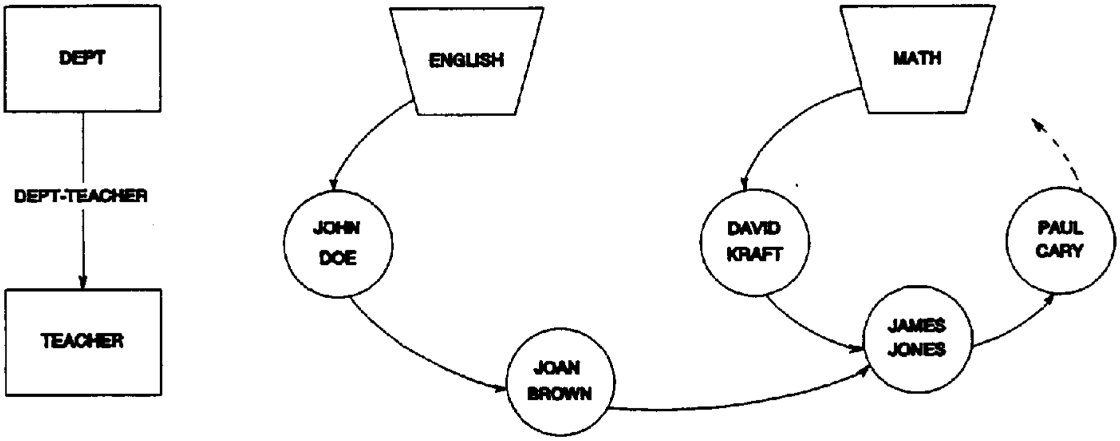


Figure 2.42: Records with Multiple Owners (Before Fix)

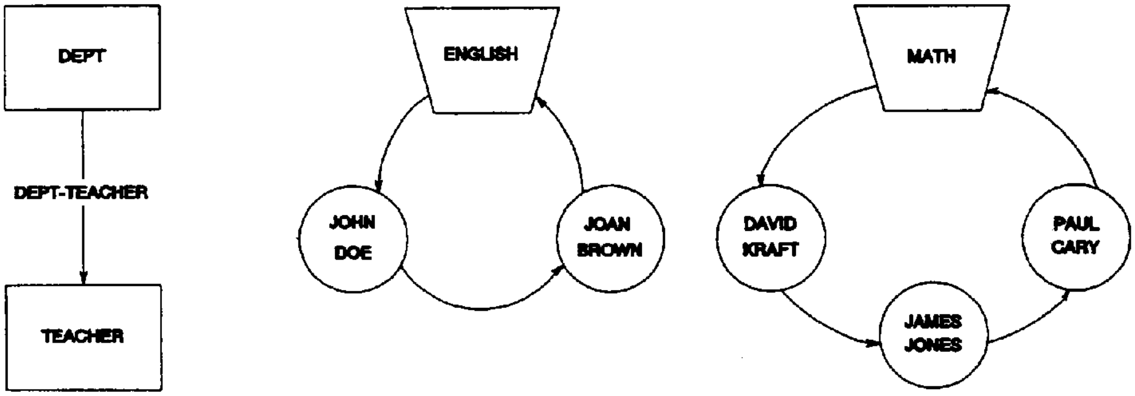


Figure 2.43: Records with Multiple Owners (After Fix)

Chapter 3. Parameters

This chapter presents CA-IDMS/DB Audit's parameter statements and options. Conventions and syntax are summarized in figure format. Detailed discussions of each parameter option follow a brief description of the parameter statements.

3.1 Notation Conventions and Syntax

Before using CA-IDMS/DB Audit parameter statements, review the following exhibits:

- Figure 3.1 Notation Conventions
- Figure 3.2 Parameter Syntax Rules
- Figure 3.3 Parameter Syntax Summary

Example	Function
PROCESS	Keywords appear in UPPERCASE. The minimum required portion of each keyword is UNDERSCORED. If a portion of or an entire keyword is not underscored, you can omit that portion or that keyword.
,AREA=*area-name*	Variables appear in lowercase italics. You must substitute an appropriate value for each variable.
[,BEGIN=*page-number*]	Brackets indicate optional clauses.
/ CRITICAL \ PROCESS= < ALL > \ NONE /	Braces enclose two or more options. You must select one of the options.
│ ,PAGES │ │ , CALCSETS │ │ ,USERSETS │ │ ,LDEL │	A pair of double bars encloses two or more options. You must select one or more of the options. Do not specify an option more than once.

Figure 3.1: Notation Conventions

Item	Rule
Order of Parameter Statements	You must enter the PROCESS statement first. You can enter all other parameter statements in any order. Within a parameter statement you can enter options in any order.
Continuing a Parameter Statement	To continue a parameter statement onto the next record, key in a trailing comma. Do not split a keyword phrase between two records.
Entering Blanks In Parameter Statements	You can enter blanks (character spaces) between keywords to improve readability in a parameter statement without affecting processing. Do not, however, embed blanks within a keyword or value field.
Entering Parameter Statements	On an 80-character input record, you must enter all parameter syntax between positions 1 through 72.
Comments	Enter an asterisk (*) in column 1, to indicate a comment.

Figure 3.2: Parameter Syntax Rules

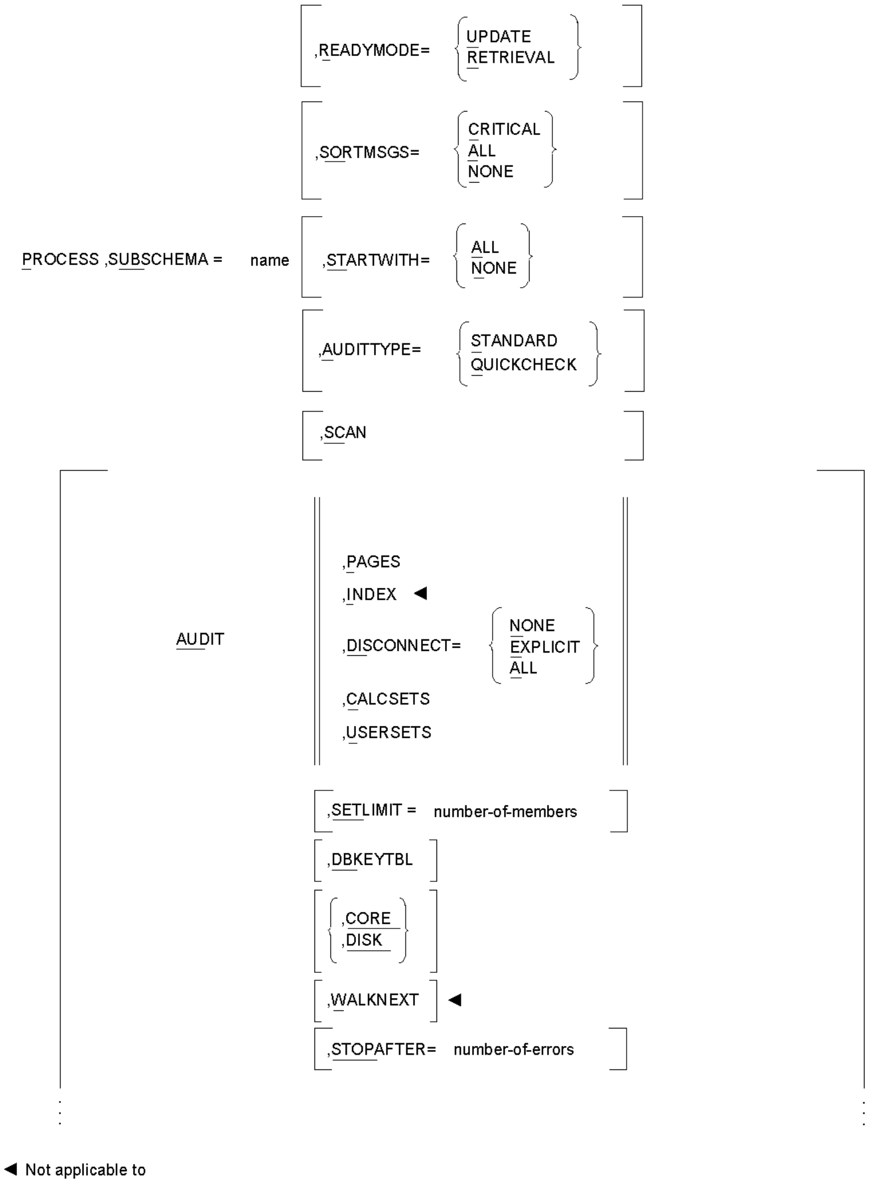
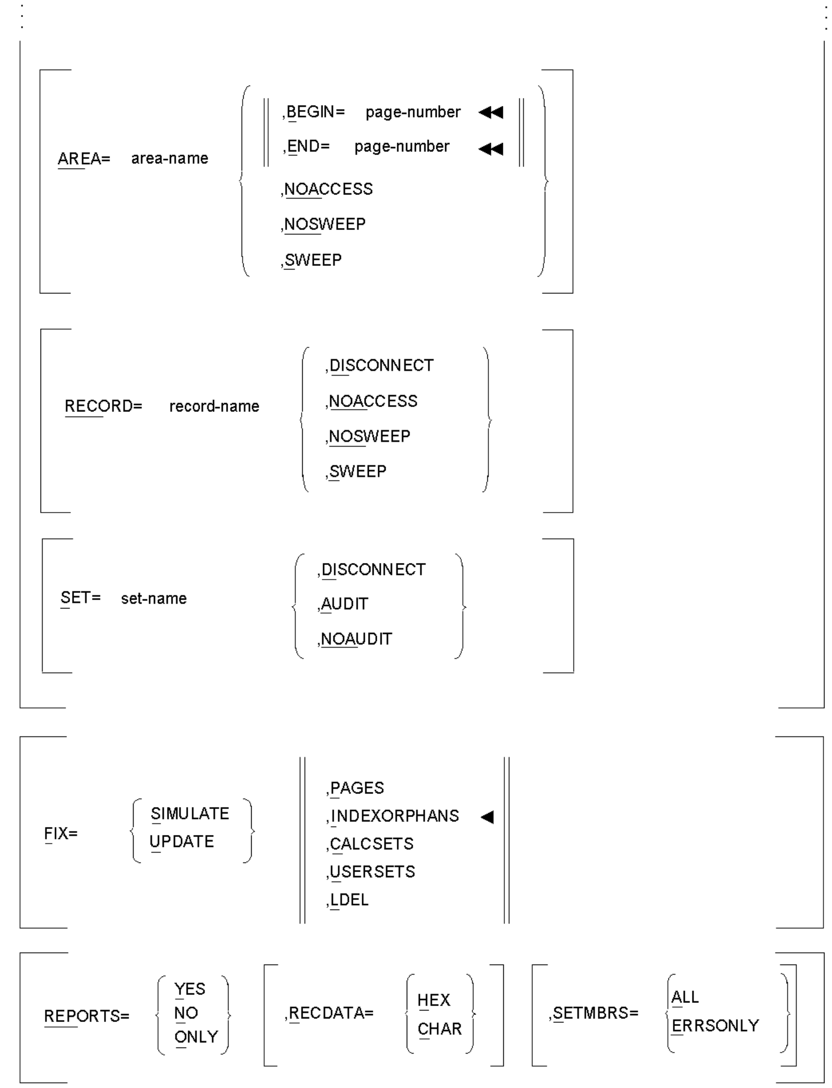


Figure 3.3a: CA-IDMS/DB Audit Parameter Syntax Summary



◀◀ Not applicable to QuickCheck if using USERSETS or

Figure 3.3b: CA-IDMS/DB Audit Parameter Syntax Summary

3.2 Parameter Statements and Options

As the syntax summary on the preceding pages displays, each CA-IDMS/DB Audit parameter statement contains a number of options. This section briefly describes the function of these statements and their options. The next section presents a detailed description of each statement and its associated options.

The function of a parameter option is the same for both Standard and QuickCheck processing, unless otherwise noted. Any differences are clearly identified and discussed separately.

3.2.1 PROCESS Statement

The PROCESS statement initiates CA-IDMS/DB Audit processing and identifies the subschema that describes the database to be audited. The PROCESS statement is the only parameter statement that is required, and it must be entered first. In addition to the PROCESS statement, you must also include at least one AUDIT, FIX, or REPORT statement with each run.

PROCESS statement options enable you to specify the READYMODE for the database and to indicate which SORT messages to display on the Audit Report. Additional options initiate inclusive or exclusive processing, specify the audit type as either Standard or QuickCheck, and enable you to check your parameters before auditing begins.

3.2.2 AUDIT Statement

The AUDIT statement specifies the scope of the audit.

AUDIT options provide exact instructions on types of errors or data structures to audit: CALC sets, user-defined sets, index sets, pages, and record or set disconnections. By specifying additional options with CALC sets or USERSETS, you can limit processing even further.

3.2.3 AREA, RECORD, and SET Statements

The AREA, RECORD, and SET statements are used in conjunction with the AUDIT statement to designate which parts of the database to process.

AREA, RECORD, and SET options direct CA-IDMS/DB Audit to include or exclude entire areas, parts of areas, and specific records or sets when auditing.

3.2.4 FIX Statement

The FIX statement enables you to request either simulated or actual correction of most types of errors. Simulated corrections allow you to examine the Integrity Analysis Report and review the projected results without actually updating the database.

FIX options specify which types of errors to correct: CALC or set errors, integrated index orphans, pointer errors, or page header and footer errors. You can apply corrections either to errors found in a previous audit, or to errors found in the current audit.

You can also use one of the FIX options to physically delete logically deleted records (LDELs).

3.2.5 REPORTS Statement

Every execution of CA-IDMS/DB Audit results in an Audit Report of processing messages and statistics. The Integrity Analysis Report is optional, however.

The REPORTS statement allows you to specify whether or not to generate an Integrity Analysis Report. This report gives details of the physical integrity errors found in the database.

REPORT options enable you to modify the format of the Integrity Analysis Report. You can display the data in either character form or in hexadecimal and character form. Other options let you control how set errors are displayed. You can also generate a report on a previous audit.

3.3 PROCESS Statement

In order to run CA-IDMS/DB Audit, you must enter the PROCESS statement first. The PROCESS statement performs these functions:

- Initiates execution of CA-IDMS/DB Audit
- Identifies the subschema
- Indicates the mode in which the database is to be readied for an audit or a fix
- Indicates the type of SORT messages to be displayed on the Audit Report
- Initiates inclusive or exclusive processing
- Specifies the audit type as either Standard or QuickCheck
- Scans your parameters for errors before auditing begins

```
                        ┌            /           \┐
                        │,READYMODE=< UPDATE     >│
                        └            \ RETRIEVAL /┘

                        ┌            / CRITICAL  \┐
                        │,SORTMSGS= < ALL        >│
                        └            \ NONE      /┘

                        ┌            /           \┐
PROCESS,SUBSCHEMA=name  │,STARTWITH=< ALL       >│
                        └            \ NONE      /┘

                        ┌            / STANDARD  \┐
                        │,AUDITTYPE=< QUICKCHECK>│
                        └            \           /┘

                        [,SCAN                    ]
```

Figure 3.4: PROCESS Statement Syntax

3.3.1 PROCESS Parameter

PROCESS, SUBSCHEMA=name

PROCESS initiates execution of CA-IDMS/DB Audit and indicates that processing options follow. CA-IDMS/DB Audit accesses and loads the subschema and its related DMCL from the load/core image library.

SUBSCHEMA identifies the subschema that describes the database to be processed. If you select an AUDIT, you can exclude specific areas, sets, or records by entering AREA, SET, or RECORD statements.

Default: There is no default value. You must enter a valid subschema name.

Rules:

- If you are auditing a multi-member set, you must include all member records of that set in the subschema; do not exclude any members from the audit. The CALC set is a multi-member set.
- All records and sets to be audited must allow the DML commands FIND, GET, MODIFY, and ERASE.
- The subschema cannot contain native VSAM records.

3.3.2 READYMODE Option

```
            /           \
,READYMODE=< UPDATE     >
            \ RETRIEVAL /
```

READYMODE specifies the mode in which CA-IDMS/DB Audit prepares the database for processing.

UPDATE readies the database in CA-IDMS's exclusive update mode. This mode prevents concurrent operations of any kind in the area. If you specify FIX=UPDATE, you must also specify READYMODE=UPDATE.

RETRIEVAL readies the database in CA-IDMS's shared retrieval mode. Do not use this mode if you are specifying FIX=UPDATE. You can, however, run other jobs against the area at the same time. If another job is updating the database, CA-IDMS/DB Audit may report errors that do not actually exist.

Default: UPDATE

3.3.3 SORTMSGS Option

```
            / CRITICAL  \
,SORTMSGS= < ALL        >
            \ NONE      /
```

SORTMSGS specifies which SORT program messages to display on the Audit Report.

CRITICAL displays only those messages that impede the execution of the SORT program.

ALL displays all SORT messages.

NONE displays no SORT messages.

Default: CRITICAL

3.3.4 STARTWITH Option

```
            /      \
,STARTWITH= < ALL  >
            \ NONE /
```

STARTWITH initiates either inclusive or exclusive processing.

ALL presumes that the entire database, as defined by the subschema, is to be audited. Optionally, you can then exclude data from processing by specifying AREA, RECORD, and SET statements.

NONE presumes that none of the database, as defined by the subschema, is to be audited. You include data for processing with AREA, RECORD, and SET statements. If you specify STARTWITH=NONE, then you must specify AREA, RECORD, or SET statements.

Default: ALL

3.3.5 AUDITTYPE Option

```
            / STANDARD  \
 ,AUDITTYPE=< QUICKCHECK>
            \           /
```

AUDITTYPE specifies the audit type as either STANDARD or QUICKCHECK.

QUICKCHECK auditing finds most common database errors with much less overhead than Standard auditing.

STANDARD auditing is the most thorough auditing method. It finds all physical integrity errors, but it also requires longer processing time.

Default: STANDARD

3.3.6 SCAN Option

```
,SCAN
```

The SCAN option gives you the opportunity to check and correct your parameter statements before processing begins. When you specify SCAN, CA-IDMS/DB Audit performs parameter validation and reports on how areas, records, and sets are going to be audited based on the options specified.

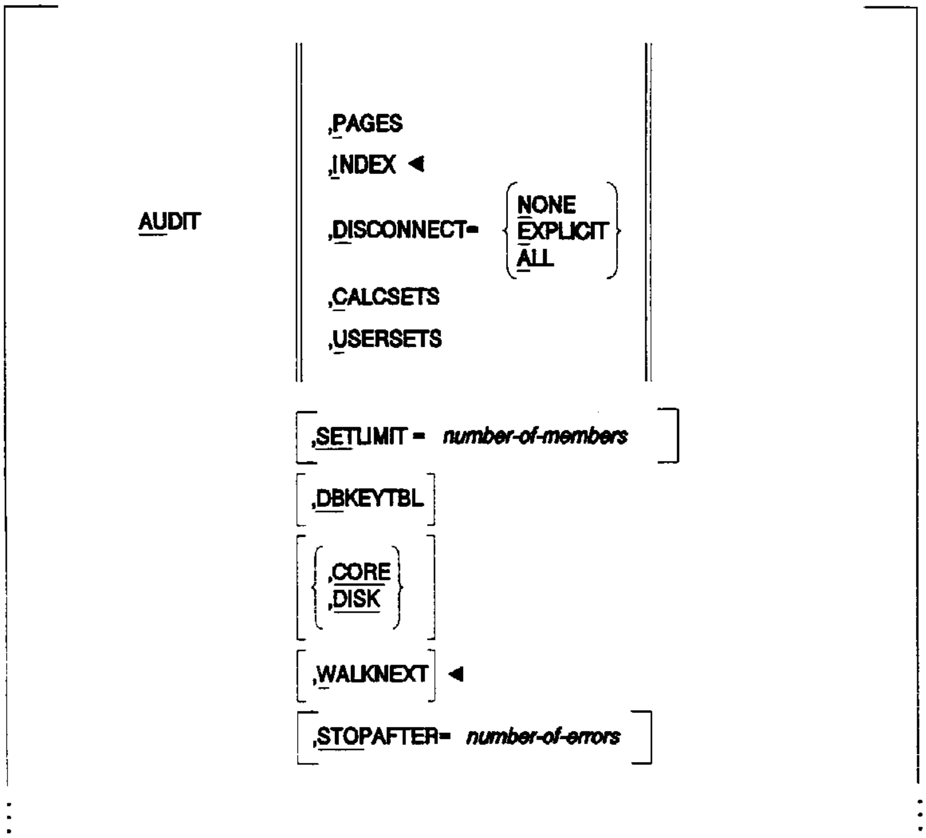
Default: There is no default value.

3.4 AUDIT Statement

The optional AUDIT statement specifies the types of physical integrity errors for CA-IDMS/DB Audit to find.

You can enter only one AUDIT statement for each execution of CA-IDMS/DB Audit. The single AUDIT statement indicates which of the six types of auditing to perform. This statement can also include a limit on the number of errors to be detected so that CA-IDMS/DB Audit terminates processing when it finds the specified number of errors.

You can use AREA, RECORD, and SET statements with the AUDIT statement to include or exclude parts of the database described in the subschema. Descriptions and syntax of these statements follow this section about the AUDIT statement.



```
AUDIT    ,PAGES
         ,INDEX ◄
         ,DISCONNECT= { NONE | EXPLICIT | ALL }
         ,CALCSETS
         ,USERSETS
         [ ,SETLIMIT = number-of-members ]
         [ ,DBKEYTBL ]
         [ { ,CORE | ,DISK } ]
         [ ,WALKNEXT ] ◄
         [ ,STOPAFTER= number-of-errors ]
```

◄ Not applicable to QuickCheck

Figure 3.5: AUDIT Statement Syntax

3.4.1 AUDIT Parameter

```
AUDIT
```

The AUDIT parameter initiates audit processing according to the audit options included in the statement. If you specify the AUDIT parameter, you must select one or more of its options: PAGES, INDEX, DISCONNECT, CALCSETS, or USERSETS.

As a result of audit processing, CA-IDMS/DB Audit writes extract records to a general extract file and, optionally, to an index extract file, depending on the parameters you specify.

3.4.2 PAGES Option

```
,PAGES
```

PAGES directs CA-IDMS/DB Audit to audit the page numbers in the header and footer of each database page encountered during area sweep processing.

3.4.3 INDEX Option

```
,INDEX
```

INDEX verifies each integrated index entry against its database record occurrence, and looks for index orphans by verifying each upper level node pointer.

INDEX does not check for next, prior, or owner (N/P/O) pointer errors in integrated index sets because this checking is performed during USERSETS processing.

If CA-IDMS/DB Audit finds pointer errors, you can fix them during the fix phase.

QuickCheck: Do not specify INDEX with QuickCheck.

3.4.4 DISCONNECT Option

```
            / NONE     \
,DISCONNECT=< EXPLICIT >
            \ ALL      /
```

DISCONNECT specifies which member records are to be searched for set disconnections. The Audit Report displays those sets that are audited for disconnections.

NONE prevents a search for set disconnections.

EXPLICIT searches for set disconnections in:

- member records of mandatory automatic sets
- sets designated DISCONNECT in a SET statement

- records designated DISCONNECT in a RECORD statement

ALL searches for set disconnections in records that are members of OPTIONAL and MANUAL sets as well as MANDATORY AUTOMATIC sets. You can use the ALL option instead of specifying DISCONNECT EXPLICIT for each individual set in the subschema.

DISCONNECT increases run time significantly because CA-IDMS/DB Audit generates a large number of I/Os to verify set disconnection.

DISCONNECT is an option in both the AUDIT, RECORD, and SET statements. In order for the DISCONNECT option of the RECORD or SET statement to have any effect, you must specify

DISCONNECT=EXPLICIT with the AUDIT statement.

QuickCheck: DISCONNECT processing identifies fully disconnected records only. Partially disconnected records are identified by QuickCheck USERSETS processing.

Standard: DISCONNECT processing identifies both partially disconnected and fully disconnected records.

Default: NONE

3.4.5 CALCSETS Option

```
,CALCSETS
```

The CALCSETS option examines the CALC set for next and prior pointer errors. For CALC set auditing to be complete, your subschema must contain a definition for all CALC records that reside in the area of the database designated for sweep processing. If your subschema does not contain definitions for all CALC sets, processing will be incomplete. If AUDITTYPE=STANDARD is specified, CALC records will also be checked to ensure they are connected to the correct CALC set.

3.4.6 USERSETS Option

```
,USERSETS
```

USERSETS examines user-defined sets and integrated index sets for next, prior, or owner (N/P/O) pointer errors. The upper level node pointer in integrated index members is not examined during USERSETS processing.

Standard: CA-IDMS/DB Audit performs LDEL processing for all sets. CA-IDMS/DB Audit walks each set from the owner to determine whether the next, prior, and owner pointers of each record are correct. If you performed a complete area sweep, you can then physically delete these logically deleted records by specifying the LDEL parameter in a FIX=UPDATE statement.

QuickCheck: CA-IDMS/DB Audit performs LDEL processing only for sets that contain N/P/O errors. If a set contains one or more N/P/O errors, CA-IDMS/DB Audit automatically creates extract records for logically deleted member records (LDELs).

3.4.7 SETLIMIT Option

```
,SETLIMIT=number-of-members
```

SETLIMIT specifies the maximum number of member records to retrieve from a set occurrence before assuming that a set loop condition exists. A set loop occurs when a member record's next db-key points back to a previous member record in the set chain.

The SETLIMIT value must be an integer from 1 through 9999999. For most efficient performance, the value should equal the maximum number of member records in any one set occurrence. Specifying too large a value causes unnecessary processing within a set when a set loop exists. Another result of specifying too large a value is unnecessarily large storage requirements for DBKEYTBL and the db-key save table.

SETLIMIT causes CA-IDMS/DB Audit to stop processing a set only when it has retrieved the number of members specified by the SETLIMIT value. For this reason, you can detect a set loop much more quickly by specifying DBKEYTBL (explained in the next section).

CA-IDMS/DB Audit does not perform any fixing if the SETLIMIT value is reached for any set occurrence.

Default: 3000

3.4.8 DBKEYTBL Option

```
,DBKEYTBL
```

DBKEYTBL directs CA-IDMS/DB Audit to build a table of the db-keys of all member records when auditing a set occurrence. CA-IDMS/DB Audit uses this table to detect a set loop. A set loop occurs when a member record's next db-key points back to a previous member in the set chain.

If you specify DBKEYTBL, CA-IDMS/DB Audit stops processing a set under either of these two circumstances:

- When CA-IDMS/DB Audit retrieves a set member whose db-key is already in the db-key table
- When there is no room to store a db-key in the table.

If you do not specify DBKEYTBL, CA-IDMS/DB Audit stops processing a set when it has retrieved the number of members indicated by the SETLIMIT value.

The maximum number of entries DBKEYTBL can hold is 120% of the SETLIMIT value. 80% of SETLIMIT is allocated to the primary portion of the table and 40% of SETLIMIT is allocated to an overflow portion of the table.

Because of the algorithm used for accessing the db-key table, there may be no room for a particular db-key in the table even though the SETLIMIT value has not been reached. Therefore, you may need to specify a SETLIMIT value slightly larger than what appears necessary.

CA-IDMS/DB Audit always allocates storage for the db-key table during the audit phase because this table is required when creating extract records for set errors. Consequently, you should always specify DBKEYTBL when auditing CALCSETS or USERSETS.

3.4.9 CORE and DISK Options

```
/ ,CORE \
< ,DISK >
\       /
```

The CORE and DISK options indicate whether to allocate core storage or disk storage for the db-key table and the db-key save table.

Default: CORE

3.4.9.1 Storage for the Db-key Table

The amount of storage used for the db-key table depends on the SETLIMIT value. The size of the db-key table is equal to 9.6 times the SETLIMIT value. For example, if the SETLIMIT value is 3000 (the default), the storage size is: 9.6 * 3000=28800 bytes.

3.4.9.2 Storage for the Db-key Save Table

The amount of storage used for the db-key save table is equal to 8 times the SETLIMIT value. For example, if the SETLIMIT value is 3000 (default), the storage required for the db-key save table is: 8 * 3000=24000 bytes.

Core storage for the db-key save table is allocated whether or not it is used. The amount of storage actually used depends on the number of next, prior, and owner errors in the part of the database being audited. Disk storage for the db-key save table is used only if N/P/O pointer errors are found.

3.4.9.3 When to Use Core Storage

Specify CORE whenever the amount of storage used by the db-key table and db-key save table will be available in the CA-IDMS/DB Audit region of your computer.

3.4.9.4 When to Use Disk Storage

If your system does not have enough core storage available, specify DISK. Very large sets, including integrated index sets, require a large SETLIMIT and, therefore, large amounts of storage for the tables. If you specify DISK, you must allocate space for two direct access files. These files are required for index processing whether or not you specify the DBKEYTBL parameter. See Chapter 5, “Operations” for more information.

3.4.9.5 Report of Allocated Storage

The amount of core storage allocated for each table is reported on the Processing Messages and Statistics section of the Audit Report. If you specify DISK, the Audit Report indicates that zero bytes of core storage have been allocated.

3.4.10 WALKNEXT Option

```
,WALKNEXT
```

WALKNEXT searches for orphan records within sets that are defined for USERSET or CALCSET processing. WALKNEXT examines each record encountered in an area sweep and walks all eligible sets of which the record is a member. If the owner does not point at the record, and none of the members point at the record, it is reported as an orphan of the set. That is, walking the set does not lead to an owner and back to the record.

The WALKNEXT parameter is usually used in a second run after a full-area audit that indicates orphan records. To determine the presence of orphan records, refer to the Audit Report, Audit Statistics. Compare the number reported for MBR RECORD OCCURRENCES plus SET DISCONNECTIONS for each set to the number reported for AREA SWEEP OCCURRENCES plus LOGICAL DELETIONS for each record that is a member of the set. Any difference between these numbers represents a possible orphan record. (If DISCONNECT was specified, the difference proves the presence of orphans for the set.)

Walknext processing causes an extraordinarily high number of I/Os because every set occurrence is walked as many times as there are member records plus once more for the owner. This type of processing increases operating time significantly.

QuickCheck: Do not specify WALKNEXT with QuickCheck. For QuickCheck processing, CA-IDMS/DB Audit searches for orphans through the USERSETS option and uses significantly less processing time.

3.4.11 STOPAFTER Option

```
,STOPAFTER=number-of-errors
```

STOPAFTER specifies the maximum number of integrity errors which CA-IDMS/DB Audit can encounter before terminating all audit processing.

When CA-IDMS/DB Audit performs more than one type of audit, it may detect a particular integrity error more than once and increment the count for that error several times. Make the STOPAFTER value large enough to allow for these duplications.

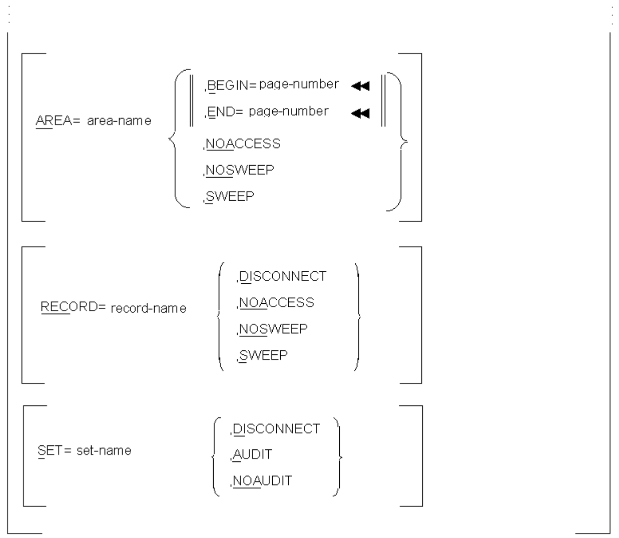
Note: If you reach the STOPAFTER value during processing, do not attempt any type of fixing because you cannot assume that CA-IDMS/DB Audit has detected all integrity errors.

Default: 999999

3.5 AREA, RECORD, and SET Statements

The AREA, RECORD, and SET statements are optional subsets of the AUDIT statement. Do not specify one of these statements unless you also specify the AUDIT statement.

The following sections describe each statement and its options.



◀◀ Not applicable to QuickCheck if using USERSETS or CALCSETS

Figure 3.6: AREA, RECORD, and SET Statement Syntax

3.6 AREA Statement

The AREA statement is used in conjunction with the PROCESS statement's STARTWITH option to include or exclude an area from being audited. You can also use the AREA statement to sweep less than the full page range of an area.

You must specify the AREA statement in conjunction with the AUDIT statement.

3.6.1 AREA Parameter

AREA=*area-name*

AREA defines the name of the area to include or exclude in processing.

3.6.2 BEGIN and END Options

,BEGIN=page-number

BEGIN specifies the page where CA-IDMS/DB Audit begins an area sweep.

Default: The first page number of the area.

,END=page-number

END specifies the page where CA-IDMS/DB Audit ends an area sweep.

Default: The last page number of the area.

SWEEP is implied with either BEGIN or END. Therefore, if you specify BEGIN or END, do not specify NOSWEEP or NOACCESS because doing so would create a conflict.

QuickCheck: You can specify BEGIN and END for QuickCheck as long as you are not also using the USERSETS or CALCSETS options. If you specify QuickCheck USERSETS or CALCSETS, do not specify BEGIN or END pages because QuickCheck relies on extracts from a complete area.

3.6.3 NOACCESS Option

,NOACCESS

The NOACCESS option omits from processing all records in the specified area. Because CA-IDMS/DB Audit does not ready the specified area, it also does not audit any sets owned by the area records, nor any sets where those records are members.

You will get an error message if you specify NOACCESS along with the DISCONNECT option in either:

- A RECORD statement for a record in the NOACCESS area

- A SET statement for a set owned by a record in the NOACCESS area

Do not specify NOACCESS in conjunction with BEGIN or END.

3.6.4 NOSWEEP Option

```
,NOSWEEP
```

NOSWEEP prevents CA-IDMS/DB Audit from processing sets owned by records in the designated area.

NOSWEEP impacts set owners only; it does not affect set members. If access is initiated from an area being swept, NOSWEEP allows each record in the area to be available.

If you specify NOSWEEP with either USERSETS or CALCSETS, NOSWEEP prevents CA-IDMS/DB Audit from processing sets owned by NOSWEEP records.

Do not specify NOSWEEP in conjunction with BEGIN or END.

3.6.5 SWEEP Option

```
,SWEEP
```

SWEEP causes the ACCESS of each record type residing in the area and the AUDIT of each set owned by records in the designated area.

SWEEP also causes the ACCESS of each set member record and, with QuickCheck, the SWEEP of each area containing these members.

Do not specify SWEEP in conjunction with BEGIN or END.

3.7 RECORD Statement

The RECORD statement is used in conjunction with the PROCESS statement's STARTWITH option to include or exclude a record type from begin audited.

You can also use the RECORD statement to report a record disconnected from a set that is not mandatory automatic. You must specify the RECORD statement in conjunction with the AUDIT statement.

3.7.1 RECORD Parameter

```
RECORD=record-name
```

RECORD specifies the name of the record type to include or exclude from auditing.

3.7.2 DISCONNECT Option

```
,DISCONNECT
```

DISCONNECT reports all records of the specified type that are disconnected from any set that is not mandatory automatic.

In order to use the RECORD,DISCONNECT option, you must specify DISCONNECT=EXPLICIT in the AUDIT statement. CA-IDMS/DB Audit automatically reports a record that is disconnected from a mandatory automatic set when you specify either DISCONNECT=EXPLICIT or DISCONNECT=ALL.

Do not specify DISCONNECT and NOACCESS together in the RECORD statement.

Specifying DISCONNECT=NONE in the AUDIT statement and DISCONNECT in the RECORD statement produces a conflict and results in an error message.

3.7.3 NOACCESS Option

```
,NOACCESS
```

NOACCESS omits the designated record from auditing. Therefore, CA-IDMS/DB Audit does not audit any sets owned by this record nor any sets where the record is a member.

Do not specify DISCONNECT and NOACCESS together in the RECORD statement.

Do not specify NOACCESS in conjunction with BEGIN or END.

3.7.4 NOSWEEP Option

,NOSWEEP

NOSWEEP prevents CA-IDMS/DB Audit from processing the specified record when sweeping the area where the record resides. Therefore, any sets owned by this record are also not audited. NOSWEEP, however, does allow each record of the specified type to be audited if processing is initiated from another area being swept.

If you specify NOSWEEP with either USERSETS or CALCSETS, NOSWEEP prevents CA-IDMS/DB Audit from processing sets owned by NOSWEEP records.

Do not specify NOSWEEP in conjunction with BEGIN or END.

3.7.5 SWEEP Option

,SWEEP

SWEEP directs CA-IDMS/DB Audit to ACCESS the designated record and SWEEP the area in which it resides. Sets owned by these records are also audited.

SWEEP also causes the ACCESS of each set member record and, with QuickCheck, the SWEEP of each area containing these members.

Do not specify SWEEP in conjunction with BEGIN or END.

3.8 SET Statement

The SET statement is used in conjunction with the PROCESS statement's STARTWITH option to include or exclude a set from being audited. You must specify the SET statement in conjunction with the AUDIT statement.

You can also use the SET statement to report a member record that is disconnected from a set occurrence that is not mandatory automatic. To do so, specify the SET statement in conjunction with the AUDIT statement's DISCONNECT=EXPLICIT option.

3.8.1 SET Parameter

```
SET=set-name
```

SET specifies the name of the set to include or exclude from auditing.

3.8.2 DISCONNECT Option

```
,DISCONNECT
```

DISCONNECT locates all records disconnected from the specified set. You must specify DISCONNECT=EXPLICIT in the AUDIT statement for the DISCONNECT option of the SET statement to have any effect. Do not specify DISCONNECT for mandatory automatic sets because records disconnected from mandatory automatic sets are automatically reported unless you specify DISCONNECT=NONE on the AUDIT statement.

Specifying DISCONNECT=NONE in the AUDIT statement and DISCONNECT in the SET statement produces a conflict and results in an error message.

3.8.3 AUDIT and NOAUDIT Options

```
,AUDIT
```

AUDIT processes the specified set for integrity errors. If you specify SET, DISCONNECT, you must also specify AUDIT. AUDIT causes the ACCESS of the owner and members of the set. AUDIT also causes SWEEP of the area containing the owner, and, with QuickCheck, SWEEP of the area containing the members.

```
,NOAUDIT
```

NOAUDIT prevents CA-IDMS/DB Audit from processing the specified set. If you specify NOAUDIT, do not specify SET, DISCONNECT.

3.9 FIX Statement

The FIX statement indicates the types of fix processing for CA-IDMS/DB Audit to simulate or perform.

Simulated fixing allows you to examine the projected results on the Integrity Analysis Report without actually updating the database.

FIX options specify which types of errors to correct: CALC or set errors, integrated index orphans, pointer errors, or page header and footer errors. You can also use the FIX statement to physically delete logically deleted records (LDELs).

You can apply corrections either to errors found in a previous audit or to errors found in the current audit. You can specify the FIX statement without an AUDIT statement only if your JCL specifies extract files that were created during a previous run of the audit phase. If CA-IDMS/DB Audit is to fix integrated index orphans that were detected in a previous run of AUDIT, you must also include the index extract file in your JCL.

You cannot specify more than one FIX statement for each run of CA-IDMS/DB Audit.

```
┌                   │ ,PAGES         │ ┐
│      / SIMULATE \ │ ,INDEXORPHANS  │ │
│FIX = < UPDATE   > │ ,CALCSETS      │ │
│      \          / │ ,USERSETS      │ │
└                   │ ,LDEL          │ ┘
```

Figure 3.7: FIX Statement Syntax

3.9.1 FIX Parameter

```
       / SIMULATE \
,FIX = < UPDATE   >
       \          /
```

The FIX parameter initiates either simulated or actual corrections of the database.

SIMULATE displays the intended corrections of detected integrity errors without actually applying the corrections to the database. You can inspect the corrections before updating the database in a subsequent audit run.

UPDATE applies corrections to integrity errors detected in the audit phase by updating the database. If you specify UPDATE in the FIX statement, you must also specify READYMODE=UPDATE in the PROCESS statement.

It is best to specify FIX=SIMULATE before using FIX=UPDATE. If the proposed corrections appear acceptable, you can then correct the detected errors by making a follow-up run using the extract files created during the audit phase.

Note: Be sure not to apply any updates to the database between running the audit and fix phases. Such updates could change the results of the fix processing or cause the fix phase to terminate abnormally.

3.9.2 PAGES Option

```
,PAGES
```

PAGES corrects errors detected within the page header and page footer. If you specify PAGES in the FIX statement, you must also specify PAGES in the audit phase.

3.9.3 INDEXORPHANS Option

```
,INDEXORPHANS
```

INDEXORPHANS corrects the upper level pointers in integrated index orphans and decreases the appropriate orphan counts. Each corrected upper level pointer points to the correct SR8 record at the next height level. CA-IDMS/DB Audit decreases the orphan counts in upper level SR8 records, starting with the upper level record designated by the old db-key (the one in error) and continuing to, but not including, the upper level record designated by the corrected upper level pointer. INDEX must have been specified in the audit phase.

If you specify INDEXORPHANS in the FIX statement, but do not specify an AUDIT statement, then you must include both the general and the index extract files as input files in the JCL.

QuickCheck: Do not specify INDEXORPHANS with QuickCheck.

3.9.4 CALCSETS Option

```
,CALCSETS
```

CALCSETS corrects integrity errors detected in the CALC set. If you specify CALCSETS in the FIX statement, you must also must have specified CALCSETS in the AUDIT statement.

3.9.5 USERSETS Option

```
,USERSETS
```

USERSETS corrects errors detected in all audited user-defined sets. In addition, USERSETS corrects next, prior, and owner pointers of integrated index sets.

3.9.6 LDEL Option

,LDEL

LDEL processing uses any logically deleted (LDEL) extracts created by USERSETS processing to physically delete LDELs if possible.

LDEL records are not deleted in sets where integrity errors are found unless you also specified USERSETS in both the AUDIT and FIX statements. LDEL records that are fully disconnected are not deleted.

If you perform a partial area audit, logically deleted records may be detected through an area sweep, but not when walking a set occurrence. In this case, logically deleted records are not physically deleted during the fix phase.

QuickCheck: Both QuickCheck and Standard USERSETS processing create LDEL record extracts for fixing. However, because QuickCheck does not walk sets that are free of error, QuickCheck USERSETS produces LDEL record extracts only for those LDELs found in user-defined sets with broken N/P/O pointers.

Figure 3.8 summarizes each FIX option, the extract file required as input for the correction, and the type of errors corrected.

FIX Options	Extract Files Required	Errors Fixed
PAGES	General	Page errors
INDEXORPHANS	General and Index	Integrated index orphans
CALCSETS	General	Integrity errors in CALC sets
USERSETS	General	N/P/O pointer errors in integrated index sets
USERSETS	General	Integrity errors in user-defined sets
LDEL	General	LDELs in user-defined sets with no integrity errors
USERSETS LDEL	General	LDELs in user-defined sets with integrity errors

Figure 3.8: Summary of FIX Options

3.10 REPORTS Statement

The REPORTS statement allows you to specify whether or not to generate an Integrity Analysis Report. You can also select the format in which record data is displayed on the reports and the format in which set related integrity errors are to be displayed.

The REPORTS statement does not influence the Audit Report, which is generated automatically with every audit run. Chapter 4, “System Output” describes the Audit Report in detail.

```
┌         / YES  \  ┌          / HEX  \┐ ┌          / ALL      \ ┐ ┐
│REPORTS=< NO    >  │,RECDATA= < CHAR >│ │,SERMBRS= < ERRSONLY > │ │
└         \ ONLY /  └          \      /┘ └          \          / ┘ ┘
```

Figure 3.9: REPORTS Statement Syntax

3.10.1 REPORTS Parameter

```
          / YES  \
REPORTS=  < NO    >
          \ ONLY /
```

The REPORTS parameter indicates whether or not you want to generate an Integrity Analysis Report.

YES causes CA-IDMS/DB Audit to generate an Integrity Analysis Report.

NO prevents CA-IDMS/DB Audit from generating an Integrity Analysis Report.

ONLY causes CA-IDMS/DB Audit to perform only the report phase, but not to perform an audit or a fix, even if you specified an AUDIT or FIX statement. ONLY generates the Integrity Analysis Report and verifies the syntax of all other parameter statements.

You can generate a report of a previous audit by specifying either YES or ONLY. To do so, include the general extract file as input to the current CA-IDMS/DB Audit job. If you specified INDEX in the previous audit, you must also include the index extract file in your JCL. Then specify a PROCESS statement without any AUDIT, FIX, or REPORTS statements.

Default: YES

3.10.2 RECDATA Options

```
          / HEX  \
,RECDATA= < CHAR >
          \      /
```

RECDATA specifies the format of set related errors on the Integrity Analysis Report. RECDATA does not affect the format of page error displays, which shows both hexadecimal and decimal form.

HEX displays data in both hexadecimal and character display formats. Db-keys are displayed in both decimal and hexadecimal formats. Three lines of data are generated for each record displayed.

CHAR displays data in character notation only (alphanumeric and special characters). Binary data is not correctly displayed in this format. Db-keys are displayed in decimal only. One line of data is generated for each record displayed.

Default: CHAR

3.10.3 SETMBRS Options

```
          / ALL      \
SERMBRS=  < ERRSONLY >
          \          /
```

SETMBRS specifies the report format of sets that contain errors.

ALL displays all of the member records in a set, even if only one record is in error.

ERRSONLY displays only the error records in a set rather than every record in the set. For each set that contains an integrity error, ERRSONLY displays:

- The owner record
- Each set member that is in error
- The prior and next member records for each member in error

Default: ALL

Chapter 4. System Output

This chapter describes the two types of reports produced by CA-IDMS/DB Audit: the Audit Report and the Integrity Analysis Report. Included in the discussion of the Integrity Analysis Report is a step-by-step example showing how to diagram a set in error.

4.1 CA-IDMS/DB Audit Reports

CA-IDMS/DB Audit produces two types of reports: an Audit Report and an Integrity Analysis Report. The format of these reports is the same whether you use QuickCheck or Standard auditing.

4.1.1 Audit Report

The Audit Report lists error, informative, and warning messages. It consists of six reporting sections that summarize all audit processing.

A variation of the Audit Report is available to users through pre-audit scanning. Rather than reporting on the processing that has already taken place, an Audit Report produced by pre-audit scanning enables you to analyze your parameters before processing begins.

4.1.2 Integrity Analysis Report

The Integrity Analysis Report consists of two report sections that detail all physical integrity errors found and show the state of the database after CA-IDMS/DB Audit has fixed those errors.

4.2 Audit Report

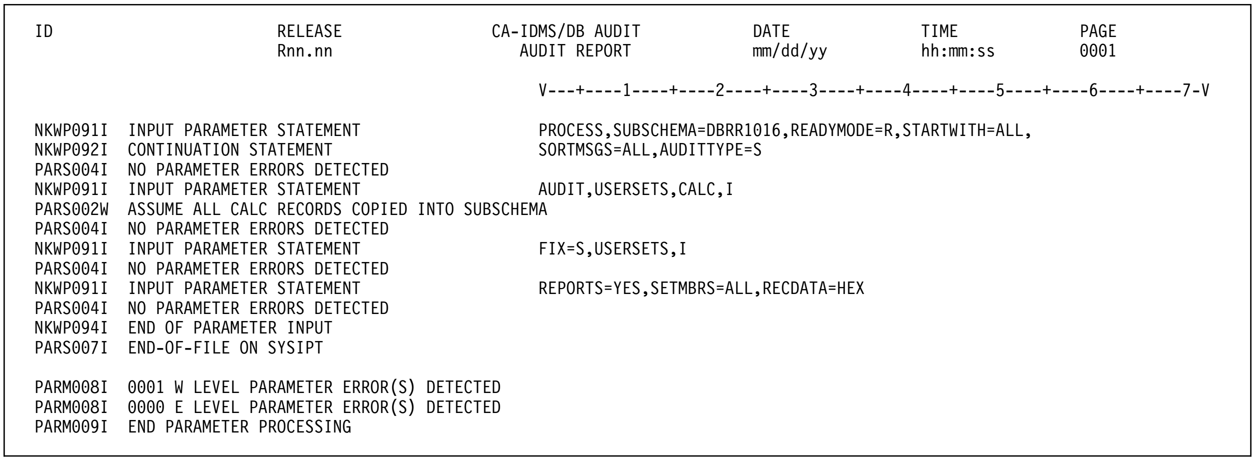
The Audit Report produces an audit trail of all processing through six report sections:

- Parameter Validation
- Parameter Processing Options
- Areas, Records, and Sets
- Runtime Status and Statistics
- Audit Statistics
- Fix Statistics

In addition to processing statistics, each section includes error, informative, and warning messages as needed.

4.2.1 Parameter Validation Report Section

The Parameter Validation report section contains the parameter statements you used to generate an audit session. It also lists pertinent messages. Figure 4.1 presents an example of this report section.



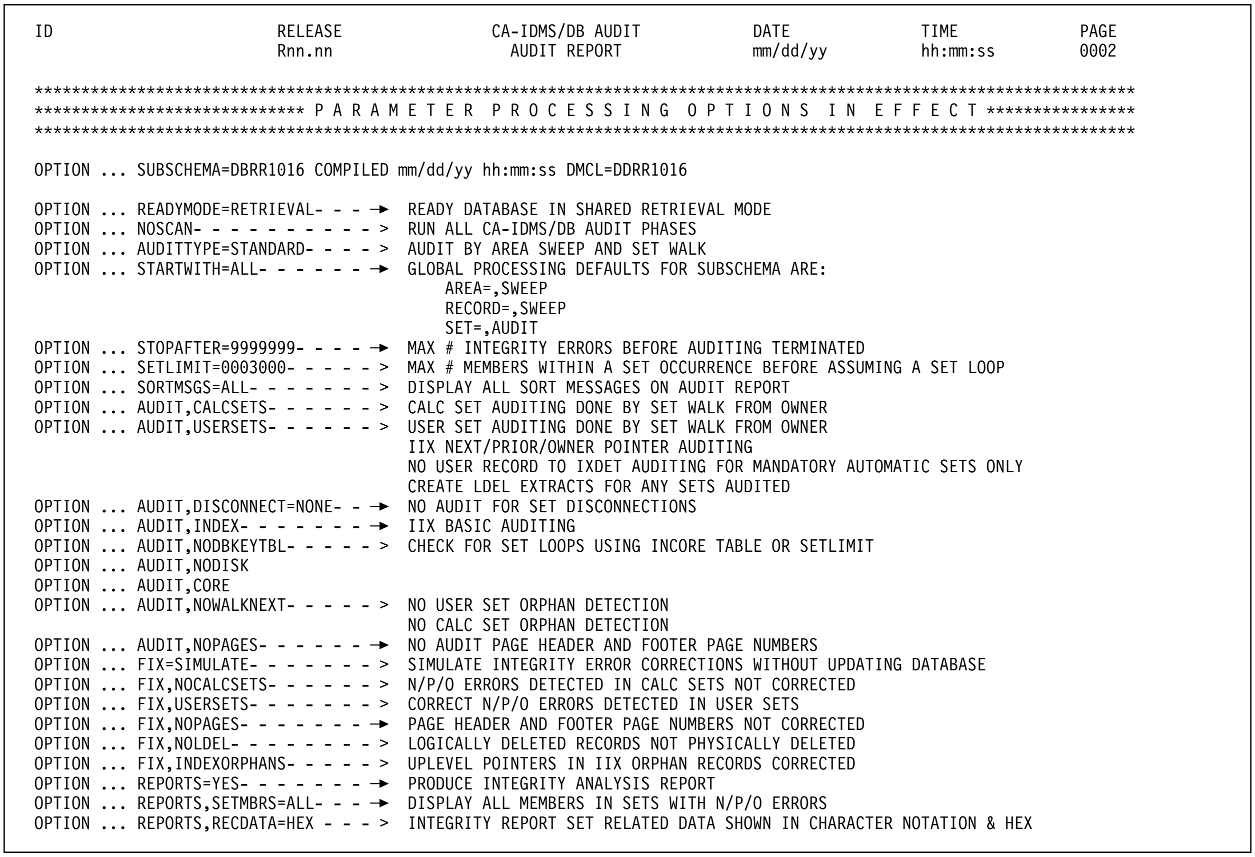
```
ID                        RELEASE              CA-IDMS/DB AUDIT            DATE            TIME             PAGE
                          Rnn.nn                  AUDIT REPORT             mm/dd/yy        hh:mm:ss         0001

                                                    V---+----1----+----2----+----3----+----4----+----5----+----6----+----7-V

NKWP091I  INPUT PARAMETER STATEMENT                 PROCESS,SUBSCHEMA=DBRR1016,READYMODE=R,STARTWITH=ALL,
NKWP092I  CONTINUATION STATEMENT                    SORTMSGS=ALL,AUDITTYPE=S
PARS004I  NO PARAMETER ERRORS DETECTED
NKWP091I  INPUT PARAMETER STATEMENT                 AUDIT,USERSETS,CALC,I
PARS002W  ASSUME ALL CALC RECORDS COPIED INTO SUBSCHEMA
PARS004I  NO PARAMETER ERRORS DETECTED
NKWP091I  INPUT PARAMETER STATEMENT                 FIX=S,USERSETS,I
PARS004I  NO PARAMETER ERRORS DETECTED
NKWP091I  INPUT PARAMETER STATEMENT                 REPORTS=YES,SETMBRS=ALL,RECDATA=HEX
PARS004I  NO PARAMETER ERRORS DETECTED
NKWP094I  END OF PARAMETER INPUT
PARS007I  END-OF-FILE ON SYSIPT

PARM008I  0001 W LEVEL PARAMETER ERROR(S) DETECTED
PARM008I  0000 E LEVEL PARAMETER ERROR(S) DETECTED
PARM009I  END PARAMETER PROCESSING
```

Figure 4.1: Sample Parameter Validation Section

4.2.2 Parameter Processing Options Section

The Parameter Processing Options report section lists all the parameter options you specified for a particular audit run. Each option is listed on a separate line, and the column on the right describes the option and its impact on processing.

Figure 4.2 presents an example of a Parameter Processing Options report section.



```
ID                        RELEASE               CA-IDMS/DB AUDIT            DATE              TIME             PAGE
                          Rnn.nn                  AUDIT REPORT              mm/dd/yy          hh:mm:ss         0002

*********************************************************************************************************************
***************************** P A R A M E T E R   P R O C E S S I N G   O P T I O N S   I N   E F F E C T ****************
*********************************************************************************************************************

OPTION ... SUBSCHEMA=DBRR1016 COMPILED mm/dd/yy hh:mm:ss DMCL=DDRR1016

OPTION ... READYMODE=RETRIEVAL- - - -→  READY DATABASE IN SHARED RETRIEVAL MODE
OPTION ... NOSCAN- - - - - - - - - - >  RUN ALL CA-IDMS/DB AUDIT PHASES
OPTION ... AUDITTYPE=STANDARD- - - - >  AUDIT BY AREA SWEEP AND SET WALK
OPTION ... STARTWITH=ALL- - - - - - -→  GLOBAL PROCESSING DEFAULTS FOR SUBSCHEMA ARE:
                                            AREA=,SWEEP
                                            RECORD=,SWEEP
                                            SET=,AUDIT
OPTION ... STOPAFTER=9999999- - - - -→  MAX # INTEGRITY ERRORS BEFORE AUDITING TERMINATED
OPTION ... SETLIMIT=0003000- - - - - >  MAX # MEMBERS WITHIN A SET OCCURRENCE BEFORE ASSUMING A SET LOOP
OPTION ... SORTMSGS=ALL- - - - - - - >  DISPLAY ALL SORT MESSAGES ON AUDIT REPORT
OPTION ... AUDIT,CALCSETS- - - - - - >  CALC SET AUDITING DONE BY SET WALK FROM OWNER
OPTION ... AUDIT,USERSETS- - - - - - >  USER SET AUDITING DONE BY SET WALK FROM OWNER
                                        IIX NEXT/PRIOR/OWNER POINTER AUDITING
                                        NO USER RECORD TO IXDET AUDITING FOR MANDATORY AUTOMATIC SETS ONLY
                                        CREATE LDEL EXTRACTS FOR ANY SETS AUDITED
OPTION ... AUDIT,DISCONNECT=NONE- - -→  NO AUDIT FOR SET DISCONNECTIONS
OPTION ... AUDIT,INDEX- - - - - - - -→  IIX BASIC AUDITING
OPTION ... AUDIT,NODBKEYTBL- - - - - >  CHECK FOR SET LOOPS USING INCORE TABLE OR SETLIMIT
OPTION ... AUDIT,NODISK
OPTION ... AUDIT,CORE
OPTION ... AUDIT,NOWALKNEXT- - - - - >  NO USER SET ORPHAN DETECTION
                                        NO CALC SET ORPHAN DETECTION
OPTION ... AUDIT,NOPAGES- - - - - - -→  NO AUDIT PAGE HEADER AND FOOTER PAGE NUMBERS
OPTION ... FIX=SIMULATE- - - - - - - >  SIMULATE INTEGRITY ERROR CORRECTIONS WITHOUT UPDATING DATABASE
OPTION ... FIX,NOCALCSETS- - - - - - >  N/P/O ERRORS DETECTED IN CALC SETS NOT CORRECTED
OPTION ... FIX,USERSETS- - - - - - - >  CORRECT N/P/O ERRORS DETECTED IN USER SETS
OPTION ... FIX,NOPAGES- - - - - - - -→  PAGE HEADER AND FOOTER PAGE NUMBERS NOT CORRECTED
OPTION ... FIX,NOLDEL- - - - - - - - >  LOGICALLY DELETED RECORDS NOT PHYSICALLY DELETED
OPTION ... FIX,INDEXORPHANS- - - - - >  UPLEVEL POINTERS IN IIX ORPHAN RECORDS CORRECTED
OPTION ... REPORTS=YES- - - - - - - -→  PRODUCE INTEGRITY ANALYSIS REPORT
OPTION ... REPORTS,SETMBRS=ALL- - - -→  DISPLAY ALL MEMBERS IN SETS WITH N/P/O ERRORS
OPTION ... REPORTS,RECDATA=HEX - - - >  INTEGRITY REPORT SET RELATED DATA SHOWN IN CHARACTER NOTATION & HEX
```

Figure 4.2: Sample Parameter Processing Options Section

4.2.3 Areas, Records, and Sets Section

The Areas, Records, and Sets report section displays only those areas, records, and sets that were audited. It also includes the area's low and high page numbers and the page numbers where the area sweep began and ended.

If you specified either the DISCONNECT or NOSWEEP parameter options, they also appear on the report opposite the appropriate area, record, or set name.

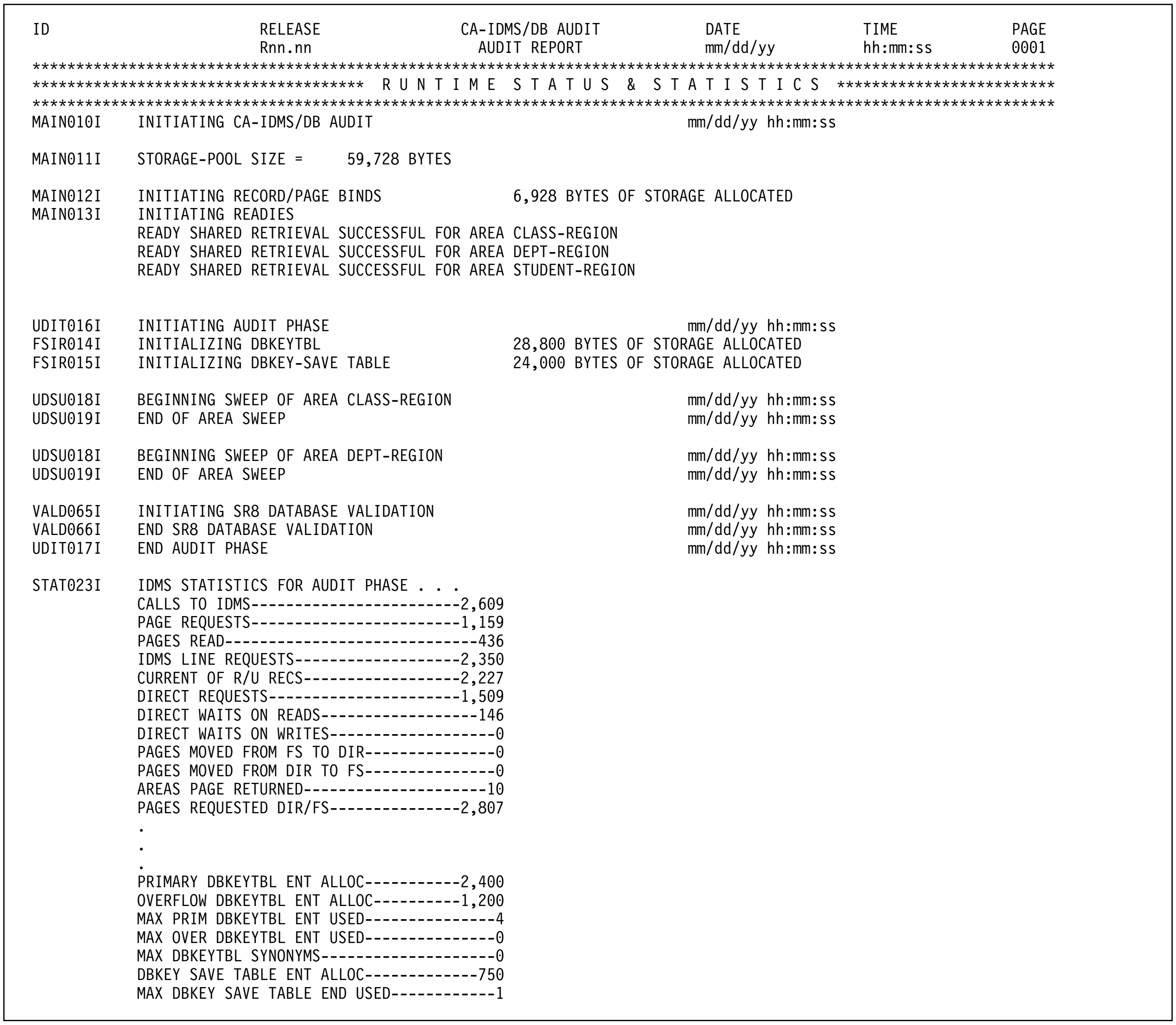
Figure 4.3 presents an example of an Area, Record, and Set report section.

```
ID                        RELEASE                CA-IDMS/DB AUDIT            DATE              TIME             PAGE
                          Rnn.nn                   AUDIT REPORT              mm/dd/yy          hh:mm:ss         0001

**************************************************************************************************************************
**************************************** A R E A S / R E C O R D S / S E T S ********************************************
**************************************************************************************************************************

AREA . . . . . .CLASS-REGION        LOPAGE(********89,001) HIPAGE(********89,030) BEGIN(********89,001) END(********89,030)
                                        READIED
                                        AREA WILL BE SWEPT
RECORD . . .  . . .ACTIVITY             ACCESSED
SET. . . . .  . . . . .ACTIVITY-TYPES   USERSETS,NOWALKNEXT
RECORD . . . . . . ARTS                 ACCESSED
RECORD . . . . . . CLASS                ACCESSED
SET. . . . . . . . . . CLASS-SCHEDULE   USERSETS,NOWALKNEXT
RECORD . . . . . . MUSIC                ACCESSED
RECORD . . . . . . SR1                  ACCESSED
SET. . . . . . . . . . CALC             CALCSETS,NOWALKNEXT

AREA . . . . . .DEPT-REGION         LOPAGE(********88,001) HIPAGE(********88,015) BEGIN(********88,001) END(********88,015)
                                        READIED
                                        AREA WILL BE SWEPT
RECORD . . . . . . DEPT                 ACCESSED
SET. . . . . . . . . . DEPT-TEACHER     USERSETS,NOWALKNEXT
SET. . . . . . . . . . DEPT-ACADEMIC    USERSETS,NOWALKNEXT
SET. . . . . . . . . . DEPT-GENERAL     USERSETS,NOWALKNEXT
SET. . . . . . . . . . IX-TCHR-LNAME    USERSETS,INDEX
RECORD . . . . . . PREREQ               ACCESSED
RECORD . . . . . . SUBJECT              ACCESSED
SET. . . . . . . . . . PREREQSFOR       USERSETS,NOWALKNEXT
SET. . . . . . . . . . PREREQSARE       USERSETS,NOWALKNEXT
SET. . . . . . . . . . SUBJECT-CLASS    USERSETS,NOWALKNEXT
RECORD . . . . . . TEACHER              ACCESSED
SET. . . . . . . . . . TEACHER-CLASS    USERSETS,NOWALKNEXT
RECORD . . . . . . SR1                  ACCESSED
SET. . . . . . . . . . CALC             CALCSETS,NOWALKNEXT

AREA.....INDEX-REGION       LOPAGE(********92,001) HIPAGE(********92,030) BEGIN(********92,001) END(********92,030)
READIED
AREA WILL BE SWEPT
RECORD......SR1             ACCESSED
SET.........CALC         CALCSETS,NOWALKNEXT
AREA.....INDX2-REGION       LOPAGE(********92,101) HIPAGE(********92,130) BEGIN(********92,101) END(********92,130)
READIED
AREA WILL BE SWEPT
RECORD......SR1             ACCESSED
SET........CALC          CALCSETS,NOWALKNEXT
RECORD......SR7             ACCESSED
SET.......IX-STUD-LNAME     USERSETS,INDEX
AREA.....INDX3-REGION       LOPAGE(********92,201) HIPAGE(********92,230) BEGIN(********92,201) END(********92,230)
READIED
AREA WILL BE SWEPT
RECORD......SR1             ACCESSED
SET........CALC          CALCSETS,NOWALKNEXT
RECORD......SR7             ACCESSED
```

Figure 4.3: Sample Areas, Records, and Sets Section

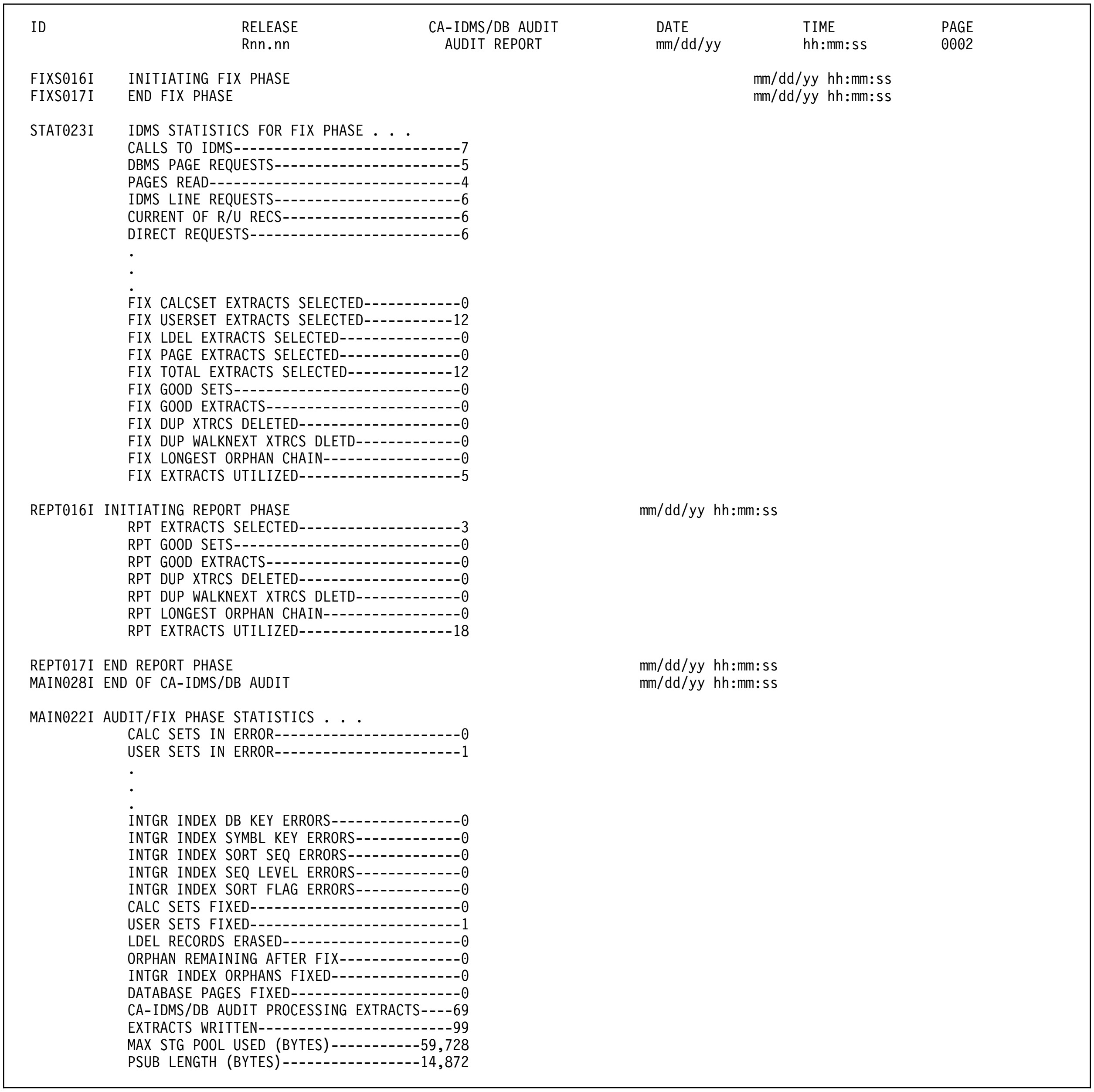
4.2.4 Runtime Status and Statistics Section

The Runtime Status and Statistics report section lists messages that detail the auditing activity through each phase. These phases are initiating CA-IDMS/DB Audit, the Audit Phase, Fix Phase, and the Report Phase.



```
ID                        RELEASE                CA-IDMS/DB AUDIT            DATE              TIME             PAGE
                          Rnn.nn                   AUDIT REPORT              mm/dd/yy          hh:mm:ss         0001
*********************************************************************************************************************
************************************** R U N T I M E   S T A T U S   &   S T A T I S T I C S ************************
*********************************************************************************************************************
MAIN010I    INITIATING CA-IDMS/DB AUDIT                                    mm/dd/yy hh:mm:ss

MAIN011I    STORAGE-POOL SIZE =     59,728 BYTES

MAIN012I    INITIATING RECORD/PAGE BINDS               6,928 BYTES OF STORAGE ALLOCATED
MAIN013I    INITIATING READIES
            READY SHARED RETRIEVAL SUCCESSFUL FOR AREA CLASS-REGION
            READY SHARED RETRIEVAL SUCCESSFUL FOR AREA DEPT-REGION
            READY SHARED RETRIEVAL SUCCESSFUL FOR AREA STUDENT-REGION


UDIT016I    INITIATING AUDIT PHASE                                         mm/dd/yy hh:mm:ss
FSIR014I    INITIALIZING DBKEYTBL                      28,800 BYTES OF STORAGE ALLOCATED
FSIR015I    INITIALIZING DBKEY-SAVE TABLE              24,000 BYTES OF STORAGE ALLOCATED

UDSU018I    BEGINNING SWEEP OF AREA CLASS-REGION                           mm/dd/yy hh:mm:ss
UDSU019I    END OF AREA SWEEP                                              mm/dd/yy hh:mm:ss

UDSU018I    BEGINNING SWEEP OF AREA DEPT-REGION                            mm/dd/yy hh:mm:ss
UDSU019I    END OF AREA SWEEP                                              mm/dd/yy hh:mm:ss

VALD065I    INITIATING SR8 DATABASE VALIDATION                             mm/dd/yy hh:mm:ss
VALD066I    END SR8 DATABASE VALIDATION                                    mm/dd/yy hh:mm:ss
UDIT017I    END AUDIT PHASE                                                mm/dd/yy hh:mm:ss

STAT023I    IDMS STATISTICS FOR AUDIT PHASE . . .
            CALLS TO IDMS------------------------2,609
            PAGE REQUESTS------------------------1,159
            PAGES READ-----------------------------436
            IDMS LINE REQUESTS-------------------2,350
            CURRENT OF R/U RECS------------------2,227
            DIRECT REQUESTS----------------------1,509
            DIRECT WAITS ON READS------------------146
            DIRECT WAITS ON WRITES-------------------0
            PAGES MOVED FROM FS TO DIR---------------0
            PAGES MOVED FROM DIR TO FS---------------0
            AREAS PAGE RETURNED---------------------10
            PAGES REQUESTED DIR/FS---------------2,807
            .
            .
            .
            PRIMARY DBKEYTBL ENT ALLOC-----------2,400
            OVERFLOW DBKEYTBL ENT ALLOC----------1,200
            MAX PRIM DBKEYTBL ENT USED---------------4
            MAX OVER DBKEYTBL ENT USED---------------0
            MAX DBKEYTBL SYNONYMS--------------------0
            DBKEY SAVE TABLE ENT ALLOC-------------750
            MAX DBKEY SAVE TABLE END USED------------1
```

Figure 4.4a: Sample Runtime Status and Statistics Section



```
ID                        RELEASE                CA-IDMS/DB AUDIT            DATE              TIME              PAGE
                          Rnn.nn                   AUDIT REPORT              mm/dd/yy          hh:mm:ss          0002

FIXS016I    INITIATING FIX PHASE                                                         mm/dd/yy hh:mm:ss
FIXS017I    END FIX PHASE                                                                mm/dd/yy hh:mm:ss

STAT023I    IDMS STATISTICS FOR FIX PHASE . . .
            CALLS TO IDMS----------------------------7
            DBMS PAGE REQUESTS-----------------------5
            PAGES READ-------------------------------4
            IDMS LINE REQUESTS-----------------------6
            CURRENT OF R/U RECS----------------------6
            DIRECT REQUESTS--------------------------6
            .
            .
            .
            FIX CALCSET EXTRACTS SELECTED------------0
            FIX USERSET EXTRACTS SELECTED-----------12
            FIX LDEL EXTRACTS SELECTED---------------0
            FIX PAGE EXTRACTS SELECTED---------------0
            FIX TOTAL EXTRACTS SELECTED-------------12
            FIX GOOD SETS----------------------------0
            FIX GOOD EXTRACTS------------------------0
            FIX DUP XTRCS DELETED--------------------0
            FIX DUP WALKNEXT XTRCS DLETD-------------0
            FIX LONGEST ORPHAN CHAIN-----------------0
            FIX EXTRACTS UTILIZED--------------------5

REPT016I INITIATING REPORT PHASE                                           mm/dd/yy hh:mm:ss
            RPT EXTRACTS SELECTED--------------------3
            RPT GOOD SETS----------------------------0
            RPT GOOD EXTRACTS------------------------0
            RPT DUP XTRCS DELETED--------------------0
            RPT DUP WALKNEXT XTRCS DLETD-------------0
            RPT LONGEST ORPHAN CHAIN-----------------0
            RPT EXTRACTS UTILIZED-------------------18

REPT017I END REPORT PHASE                                                  mm/dd/yy hh:mm:ss
MAIN028I END OF CA-IDMS/DB AUDIT                                           mm/dd/yy hh:mm:ss

MAIN022I AUDIT/FIX PHASE STATISTICS . . .
            CALC SETS IN ERROR-----------------------0
            USER SETS IN ERROR-----------------------1
            .
            .
            .
            INTGR INDEX DB KEY ERRORS----------------0
            INTGR INDEX SYMBL KEY ERRORS-------------0
            INTGR INDEX SORT SEQ ERRORS--------------0
            INTGR INDEX SEQ LEVEL ERRORS-------------0
            INTGR INDEX SORT FLAG ERRORS-------------0
            CALC SETS FIXED--------------------------0
            USER SETS FIXED--------------------------1
            LDEL RECORDS ERASED----------------------0
            ORPHAN REMAINING AFTER FIX---------------0
            INTGR INDEX ORPHANS FIXED----------------0
            DATABASE PAGES FIXED---------------------0
            CA-IDMS/DB AUDIT PROCESSING EXTRACTS----69
            EXTRACTS WRITTEN------------------------99
            MAX STG POOL USED (BYTES)-----------59,728
            PSUB LENGTH (BYTES)-----------------14,872
```

Figure 4.4b: Sample Runtime Status and Statistics Section

4.2.5 Audit Statistics Section

The Audit Statistics report section lists the areas, records, and sets that have been swept. It also identifies which pages have been swept and whether or not it detected any logical deletions or set disconnections.

Figure 4.5 presents an example of an Audit Statistics report section.

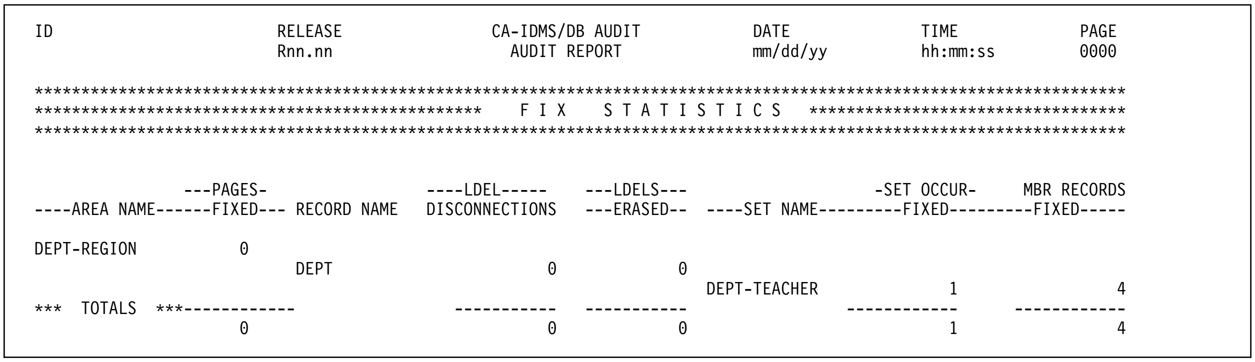
```
ID                        RELEASE                CA-IDMS/DB AUDIT            DATE              TIME             PAGE
                          Rnn.nn                   AUDIT REPORT              mm/dd/yy          hh:mm:ss         000
*********************************************************************************************************************
********************************************    A U D I T   S T A T I S T I C S   ***********************************
*********************************************************************************************************************

                 --PAGES---                  -AREA SWEEP   --LOGICAL--                  -MBR RECORD    ------SET-----
----AREA NAME------SWEPT--- ---RECORD NAME-- OCCURRENCES   -DELETIONS-  ----SET NAME----OCCURRENCES    DISCONNECTIONS

CLASS-REGION         30     ACTIVITY                   3             0  ACTIVITY-TYPES            9                 0
                            ARTS                       3             0
                            CLASS                     29             0
                                                                        CLASS-SCHEDULE           98                 0
                            MUSIC                      3             0
                            SR1                       30             0
                                                                        CALC                      3                 0
DEPT-REGION          15
                            DEPT                       6             0
                                                                        DEPT-TEACHER             14                 0
                                                                        DEPT-ACADEMIC           132                 0
                                                                        DEPT-GENERAL            132                 0
                                                                        IX-TCHR-LNAME            24    SR8'S
                            PREREQ                     7             0
                            SUBJECT                  132             6
                            PREREQSFOR                17             0
                            PREREQSARE                17             0
                            SUBJECT-CLASS             29             0
                            TEACHER                   35             0
                                                                        TEACHER-CLASS            29                 0
                            SR1                       15             0
                                                                        CALC                    167                 0
INDEX-REGION         30
                            SR1                       30             0
                      .                                                 CALC                      0                 0
                      .
                      .

MISC1-REGION         20
                            SR1                       20             0
                                                                        CALC                      0                 0
MISC2-REGION         20
                            SR1                       20             0
                                                                        CALC                      0                 0
STUDENT-REGION       80
                            DREPORT                    2             0
                            GREPORT                    3             0
                            SCHEDULE                  98             0
                            SPORTS                     3             0
                            STUDENT                   51             0
                                                                        STUDENT-SCHEDULE         51    SR8'S
                                                                        STUDENT-REPORTS           5                 0
                                                                        STUDENT-ACTIVITY          3                 0
                            SR1                       80             0
                                                                        CALC                     54                 0
***  TOTALS  ***-----------                 ------------   -----------                 ------------       -----------
                    290                              718             6                          916                 0

                            --IIX SET NAME--             INDEX ENTRIES           MEMBER DISCONNECTS

                            IX-TCHR-LNAME                           35                            0
                            STUDENT-SCHEDULE                        98                            0
                            IX-STUD-LNAME                           51                            0
                            IX-SUBJ-NAME                           126                            6
                            IX-DEPT-NAME                             6                            0
                            ***  TOTALS  ***             -------------             ----------------
                                                                   316                            6
```

Figure 4.5: Sample Audit Statistics Section

4.2.6 Fix Statistics Section

The Fix Statistics report section lists only those areas, records, and sets that CA-IDMS/DB Audit fixed, regardless of which areas, records, and sets it audited.

Figure 4.6 presents an example of a Fix Statistics report section. Fix Statistics are not displayed if the Integrity Analysis Report is suppressed.



```
ID                        RELEASE                CA-IDMS/DB AUDIT            DATE              TIME             PAGE
                          Rnn.nn                   AUDIT REPORT              mm/dd/yy          hh:mm:ss         0000

*********************************************************************************************************************
************************************************    F I X    S T A T I S T I C S   **********************************
*********************************************************************************************************************


                ---PAGES-                 ----LDEL-----    ---LDELS---                    -SET OCCUR-     MBR RECORDS
----AREA NAME------FIXED--- RECORD NAME   DISCONNECTIONS   ---ERASED--  ----SET NAME---------FIXED---------FIXED-----

DEPT-REGION           0
                            DEPT                       0             0
                                                                        DEPT-TEACHER              1                 4
***  TOTALS  ***------------                 -----------   -----------                 ------------      ------------
                      0                                0             0                            1                 4
```

Figure 4.6: Sample Fix Statistics Section

4.3 Pre-Audit Scan

CA-IDMS/DB Audit enables you to preview your audit instructions before processing begins. Pre-audit scanning analyzes the parameters you specify and provides you with the following feedback:

- Complete parameter validation
- Short description of processing options in effect
- Parameter-only processing and reporting
- Identification and matching of subschema areas, records, and sets with the processing options that affect them

If you need to correct syntax errors or modify your parameters in any way, you can do so before execution takes place.

To initiate pre-audit scanning, specify the SCAN option with the PROCESS statement.

The following section compares the sequence of events and the output if you generate an Audit Report with or without pre-audit scanning.

4.3.1 Processing Without SCAN

When you initiate audit processing without the SCAN option, the sequence of events is:

1. User specifies parameters without the SCAN option
2. CA-IDMS/DB Audit produces three Audit Report sections as output:
   - Parameter Validation
   - Processing Options
   - Areas/Records/Sets
3. CA-IDMS/DB Audit begins audit processing and simultaneously produces the Audit Report section Runtime Status/Statistics
4. After processing ends, CA-IDMS/DB Audit produces the final two Audit Report sections:
   - Audit Statistics
   - Fix Statistics

4.3.2 Processing With SCAN

The major advantage of processing with the SCAN option is that you have an opportunity to review your parameters and make corrections before audit execution takes place. The SCAN option causes events to take place in the following sequence:

1. User specifies parameters with the SCAN option
2. CA-IDMS/DB Audit produces three Audit Report sections as output:
   - Parameter Validation
   - Processing Options
   - Areas/Records/Sets
3. User modifies parameters as needed
4. User initiates parameter processing again with SCAN to verify that all parameter corrections have been properly made
5. If CA-IDMS/DB Audit detects no further errors, user imitates actual audit processing by specifying the same parameters without SCAN

Figure 4.7 presents a simplified example of SCAN output. The first report section, Parameter Validation, lists the parameters you specified and indicates if CA-IDMS/DB Audit detects any parameter errors. The Processing Options section individually identifies each parameter option in the left column and provides a brief description of that option on the right. Finally, the Areas/Records/Sets section describes the impact of all parameters against the designated parts of the subschema.

```
********************************************************************************************************
                                                  PARAMETER VALIDATION
********************************************************************************************************

PROCESS,SCAN,AUDITTYPE=QUICKCHECK,STARTWITH=ALL...

AUDIT,USERSETS...
```

```
***********************************************************************************************************
                                                 PARAMETER PROCESSING OPTIONS
***********************************************************************************************************


OPTION...SCAN------------------------------------►      CHECK PARAMETER ONLY

OPTION...AUDITTYPE=QUICKCHECK   ─────────────────►      AUDIT TYPE AREA SWEEP AND SORT

OPTION...STARTWITH=ALL---------------------------►      GLOBAL PROCESSING DEFAULTS FOR SUBSCHEMA ARE:
                                                           AREA=SWEEP
                                                           RECORD=,SWEEP
                                                           SET=,AUDIT

OPTION...AUDIT,USERSETS--------------------------►      USER SET AUDITING DONE BY QUICKCHECK
                                                           SORT AND MATCH

                                                        IIX NEXT/PRIOR/OWNER POINTER AUDIT
                                                        NO USER RECORD TO IXDET AUDITING
                                                           FOR MANDATORY AUTOMATIC SETS ONLY
                                                        CREATE LDEL EXTRACKTS FOR SETS IN ERROR ONLY

OPTION...NOINDEX---------------------------------►      NO IIX BASIC AUDITING

OPTION...NOWALKNEXT------------------------------►      USER SET ORPHAN DETECTION DONE BY
                                                           QUICKCHECK SORT AND MATCH

                                                        NO CALC SET ORPHAN DETECTION
```

```
***********************************************************************************************************
                                                      AREAS/RECORDS/SETS
***********************************************************************************************************

AREA......DEPT-REGION                                      ...
                                                           READIED
                                                           AREA WILL BE SWEPT

RECORD.......DEPT                                          ACCESSED

SET.........................DEPT-TEACHER                   USERSETS,NOWALKNEXT

SET.........................IX-TCHR-LNAME                  USERSETS,NOINDEX

RECORD.......TEACHER                                       ACCESSED
```

Figure 4.7: Sample SCAN Output

4.4 Integrity Analysis Report

The second report type produced by CA-IDMS/DB Audit is the Integrity Analysis Report. This report consists of two report sections:

- Before Image Report Section
- After Image Report Section

The **Before Image Report Section** lists all physical integrity errors found by CA-IDMS/DB Audit during the audit phase.

Before images are the extracts or record identifiers (db-key pointers, next, prior, or owner (N/P/O) pointers, sort keys) that appear on the Integrity Analysis Report.

The **After Image Report Section** shows the state of the database after CA-IDMS/DB Audit has fixed all the physical integrity errors it can fix.

After images also appear on the Integrity Analysis Report and are the same record identifiers (db-key pointers, N/P/O pointers, sort keys) in fixed format--either the way CA-IDMS/DB Audit will fix them (simulate) or the way in which CA-IDMS/DB Audit has already fixed them (update).

To generate the After Image Report without actually performing the fixes, specify SIMULATE in the FIX statement. You can have CA-IDMS/DB Audit generate the Simulated After Image Report in one run, and perform the fixes in another separate run without re-auditing the database. To execute a separate fix run, you must save the extract file (or files) created by the audit phase to use later in the fix phase.

4.4.1 Diagraming Errors

The next several pages of this section contain step-by-step instructions that show how to draw a diagram of a set in error by using the information shown on a Before Image Report.

Following the discussion of diagraming errors, there are sample reports showing how various types of errors are reported. The types of errors listed in Figure 4.8 are illustrated in the exhibits indicated.

Type of Error	**To find this error specify:**	**Report Illustrated in Figure:**	
		Before Image	**After Image**
Pointer	USERSETS CALCSETS	4.11	
Set Integrity Miscellaneous	USERSETS CALCSETS	4.12	4.13

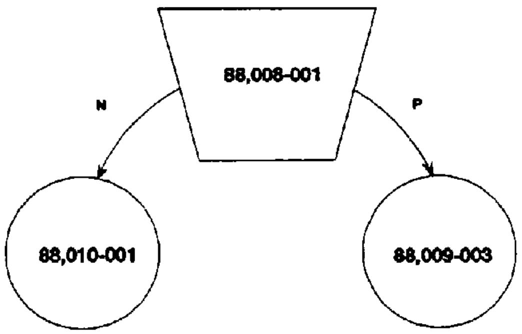
Set Disconnections	DISCONNECT = ALL
	DISCONNECT = EXPLICIT
		4.14	

IIX Pointers	INDEX	4.15	4.16
IIX Orphans	INDEX	4.17	4.18
IIX Integrity Miscellaneous	INDEX	4.19, 4.20	
From/Thru Page Range	USERSETS CALCSETS	4.21	
Orphans of sets that are not IIX	USERSETS, WALKNEXT CALCSETS, WALKNEXT AUDITTYPE = QUICKCHECK	4.22	
Fragmented Records	USERSETS CALCSETS	4.23	
Compressed Records	USERSETS CALCSETS	Not Illustrated	Not Illustrated
LDEL	USERSETS		4.24
Header Footer	PAGES	4.25	4.26

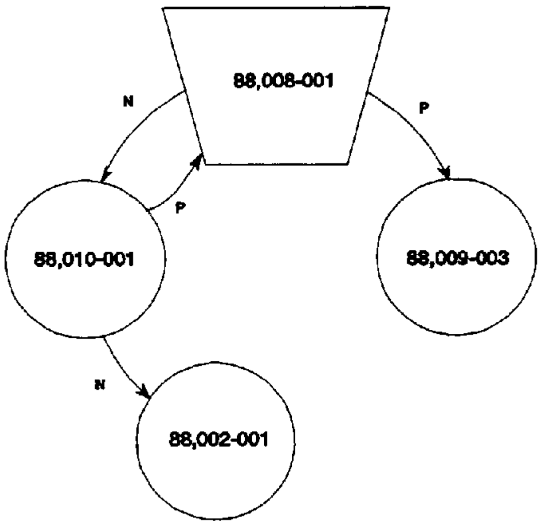
Figure 4.8: Integrity Analysis Report Illustrations

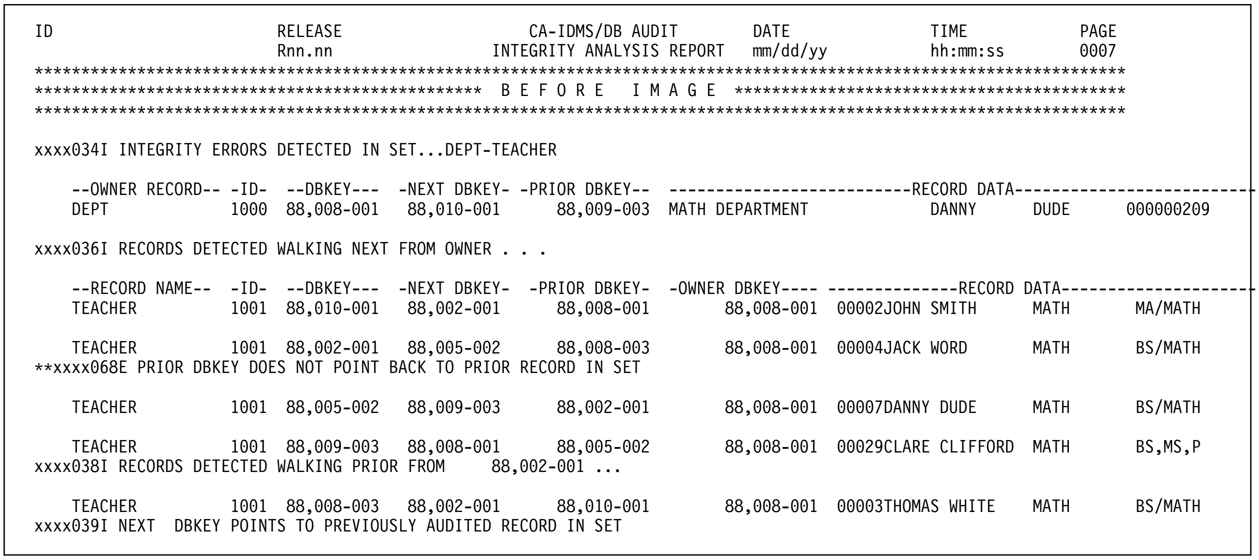
In order to understand better how CA-IDMS/DB Audit detected certain errors, you can draw a picture of the set in error by using the information contained in the Before Image Report. For the following example, see the Before Image Report in Figure 4.9, and follow the step-by-step procedure for drawing the set in error.

- **Step 1**. Message xxxx034I indicates integrity errors in the DEPT-TEACHER set occurrence.
- **Step 2**. Starting with the OWNER record (88,008-001) the NEXT (first) member is listed as 88,010-001 and the PRIOR (last) member is 88,009-003.



- **Step 3**. Message xxxx036I indicates that CA-IDMS/DB Audit will walk next from the OWNER record.
- **Step 4**. The NEXT record (88,010-001) points to 88,002-001 as the NEXT member in the set and points back to the OWNER record (88,008-001).



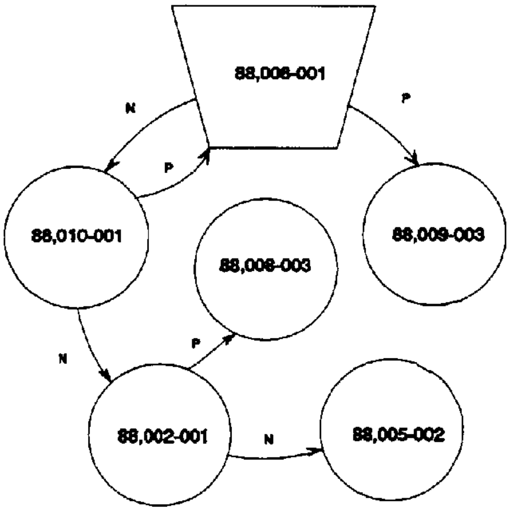


```
ID                        RELEASE                    CA-IDMS/DB AUDIT        DATE              TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy          hh:mm:ss        0007
***********************************************************************************************************************
************************************************  B E F O R E   I M A G E  ********************************************
***********************************************************************************************************************

xxxx034I INTEGRITY ERRORS DETECTED IN SET...DEPT-TEACHER

    --OWNER RECORD-- -ID-  --DBKEY---  -NEXT DBKEY- -PRIOR DBKEY--  --------------------------RECORD DATA-------------------------
    DEPT            1000  88,008-001   88,010-001      88,009-003  MATH DEPARTMENT             DANNY      DUDE      000000209

xxxx036I RECORDS DETECTED WALKING NEXT FROM OWNER . . .

    --RECORD NAME--  -ID-  --DBKEY---  -NEXT DBKEY-  -PRIOR DBKEY-  -OWNER DBKEY---- --------------RECORD DATA--------------------
    TEACHER         1001  88,010-001   88,002-001      88,008-001        88,008-001  00002JOHN SMITH      MATH       MA/MATH

    TEACHER         1001  88,002-001   88,005-002      88,008-003        88,008-001  00004JACK WORD       MATH       BS/MATH
**xxxx068E PRIOR DBKEY DOES NOT POINT BACK TO PRIOR RECORD IN SET

    TEACHER         1001  88,005-002   88,009-003      88,002-001        88,008-001  00007DANNY DUDE      MATH       BS/MATH

    TEACHER         1001  88,009-003   88,008-001      88,005-002        88,008-001  00029CLARE CLIFFORD  MATH       BS,MS,P
xxxx038I RECORDS DETECTED WALKING PRIOR FROM    88,002-001 ...

    TEACHER         1001  88,008-003   88,002-001      88,010-001        88,008-001  00003THOMAS WHITE    MATH       BS/MATH
xxxx039I NEXT  DBKEY POINTS TO PREVIOUSLY AUDITED RECORD IN SET
```

Figure 4.9:: Before Image Report

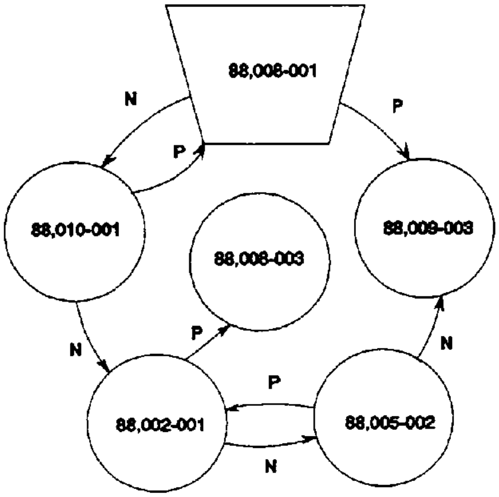
Diagraming Errors--Before Image Report

- **Step 5**. Member 88,002-001 points to 88,005-002 as the NEXT member in the set and points back to record 88,008-003.

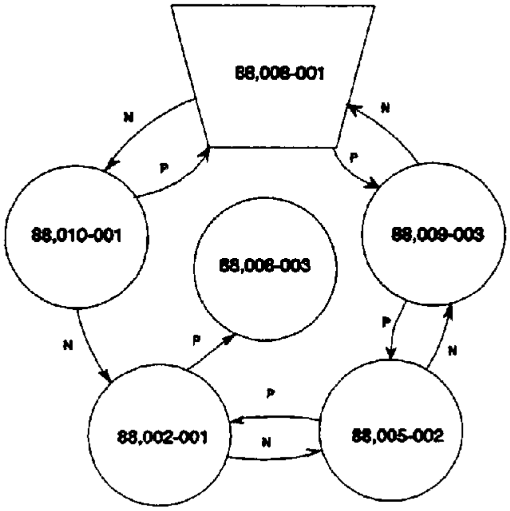


- **Step 6**. Error message xxxx068E indicates that the previous step is in error. The error is that record 88,002-001's PRIOR pointer does not point back to record 88,010-001. At this point, CA-IDMS/DB Audit saves db-key 88,002-001 for later processing. See Chapter 2, “Concepts.”

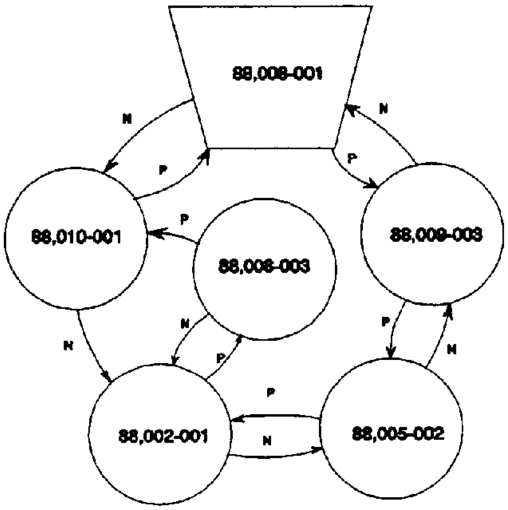
- **Step 7**. Continuing with the next member in the set (88,005-002), it points to 88,009-003 as the NEXT member in the set and points back to record 88,002-001.



- **Step 8**. Member 88,009-003 points to the OWNER record (88,008-001) as the NEXT record in the set, and points back to record 88,005-002. At this point, CA-IDMS/DB Audit stops walking the set in the next direction.



- **Step 9**. Message xxxx038I indicates that CA-IDMS/DB Audit has retrieved db-key 88,002-001 and will start walking the set in the prior direction. The first record found is 88,008-003. This record points to record 88,002-001 as the NEXT record in the set, and points to record 88,010-001 as the PRIOR record in the set.

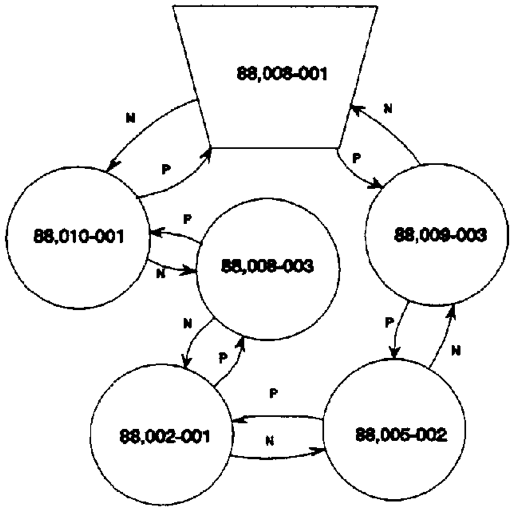


- **Step 10**. Message xxxx039I indicates that this is as far as CA-IDMS/DB Audit can process this set occurrence because record 88,008-003 points to a record that was already processed.

  This step completes the diagraming of the indicated error.

  In Figure 4.10 is a Simulated After Image Report that shows how CA-IDMS/DB Audit would fix the error that was diagramed.

  The following diagram shows how the fixed set occurrence will look. The physical integrity error is fixed. The logical integrity should now be examined. See 2.8.4, “Logical Integrity” for an example of a logical integrity error.



```
ID                        RELEASE                CA-IDMS/DB AUDIT        DATE              TIME            PAGE
                          Rnn.nn             INTEGRITY ANALYSIS REPORT   mm/dd/yy          hh:mm:ss        0019
*******************************************************************************************************************
*******************************************  S I M U L A T E D   A F T E R   I M A G E  ***************************
*******************************************************************************************************************

xxxx035I FIXING PERFORMED ON SET...DEPT-TEACHER

    --OWNER RECORD-- -ID-  -DBKEY---- -NEXT DBKEY-   -PRIOR DBKEY-    --------------------------------RECORD DATA--------------

    DEPT             1000  88,008-001   88,010-001      88,009-003    MATH DEPARTMENT        DANNY         DUDE    0000000208

    ---RECORD NAME-- -ID-  --DBKEY--- -NEXT DBKEY-  -PRIOR DBKEY-     -OWNER DBKEY--  ------------RECORD DATA------------------

    TEACHER          1001  88,010-001   88,008-003      88,008-001        88,008-001  00002JOHN   SMITH     MATH   MA/MATH
    TEACHER          1001  88,008-003   88,002-001      88,010-001        88,008-001  00003THOMAS  WHITE    MATH   BS/MATH
    TEACHER          1001  88,002-001   88,005-002      88,008-003        88,008-001  00004JACK    WORD     MATH   BS/MATH
    TEACHER          1001  88,005-002   88,009-003      88,002-001        88,008-001  00007DANNY   DUDE     MATH   BS/MATH
    TEACHER          1001  88,009-003   88,008-001      88,005-002        88,008-00100029CLARE     CLIFFORD MATH   BS,MS,P
```

Figure 4.10: Simulated After Image Report

4.4.2 Pointer Errors

An example of a record that contains an invalid prior db-key is shown in Figure 4.11. Records displayed on this section of the report are grouped by set name.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates the set found to contain pointer errors.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

NEXT DBKEY - Next pointer.

PRIOR DBKEY - Prior pointer (if present).

OWNER DBKEY - Owner pointer (if present).

RECORD DATA - First 45 characters of data.

Error messages - Indicate the types of errors detected.

```
ID                        RELEASE                      CA-IDMS/DB AUDIT      DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0009
**********************************************************************************************************************
**************************************************  B E F O R E   I M A G E  *****************************************
**********************************************************************************************************************
xxxx034I INTEGRITY ERRORS DETECTED IN SET...ACTIVITY-TYPES
    --OWNER RECORD-- -ID-- --DBKEY---  -NEXT DBKEY-  -PRIOR DBKEY-       --------------------RECORD DATA-------------
    ACTIVITY         8111  89,010-001   91,003-014      91,048-001       ACTIVITY03TRACK AND FIELD
xxxx036I RECORDS DETECTED WALKING NEXT FROM OWNER . . .

    ---RECORD NAME-- -ID-- --DBKEY---  -NEXT DBKEY-  -PRIOR DBKEY-     -OWNER DBKEY-    -----------------RECORD DATA----------
    SPORTS           8112  91,003-014   91,019-001      89,010-001        89,010-001    BROADJUMP1WOMENS BROAD JUMP
    SPORTS           8112  91,019-001   91,048-001   *,***,***-***        89,010-001    BROADJUMP2MENS BROAD JUMP
**xxxx060E PRIOR DBKEY IS NULL
    SPORTS           8112  91,048-001   89,010-001      91,019-001        89,010-001    TRACK  101HURDLES
xxxx037I RECORDS DETECTED WALKING PRIOR FROM OWNER . . .
    SPORTS           8112  91,048-001   89,010-001      91,019-001        89,010-001    TRACK  101HURDLES
    SPORTS           8112  91,019-001   91,048-001   *,***,***-***        89,010-001    BROADJUMP2MENS BROAD JUMP
**xxxx060E PRIOR DBKEY IS NULL
```

Figure 4.11: Pointer Errors

4.4.3 Set Errors -- Before Image Report

Examples of the Before Image Report fields for set errors are shown in Figure 4.12.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates the set found to contain pointer errors.

Line 7 - OWNER record information:

- **OWNER RECORD** - Name of the owner.
- **ID** - Record identifier.
- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior pointer (if present).
- **RECORD DATA** - First 59 characters of data.

Line 10 - Member record information:

- **RECORD NAME** - Name of the record.
- **ID** - Record identifier.
- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior db-key (if present).
- **OWNER DBKEY** - Owner pointer (if present).
- **RECORD DATA** - First 45 characters of data.

Error messages - Indicate the types of errors detected.

```
ID                        RELEASE                      CA-IDMS/DB AUDIT     DATE              TIME            PAGE
                          Rnn.nn                INTEGRITY ANALYSIS REPORT   mm/dd/yy          hh:mm:ss        0007
*********************************************************************************************************************
**************************************************  B E F O R E   I M A G E  ****************************************
*********************************************************************************************************************
xxxx034I INTEGRITY ERRORS DETECTED IN SET...PREREQSFOR
    --OWNER RECORD-- -ID-  --DBKEY---  -NEXT DBKEY-  -PRIOR DBKEY-     -------------RECORD DATA-----------------------
    SUBJECT          1002  88,002-009   88,002-012      88,002-012     MU201ADV MUSIC APPREC 0304 MUSIC MASTERS 010696

xxxx036I RECORDS DETECTED WALKING NEXT FROM OWNER . . .
    ---RECORD NAME-- -ID-  --DBKEY---  -NEXT DBKEY-  -PRIOR DBKEY-     -OWNER DBKEY-       ------RECORD DATA----------
    PREREQ           8103  88,002-012   88,002-012      88,002-009        88,002-009
**xxxx071E SET LOOP DETECTED
xxxx039I NEXT  DBKEY POINTS TO PREVIOUSLY AUDITED RECORD IN SET

xxxx037I RECORDS DETECTED WALKING PRIOR FROM OWNER . . .
    PREREQ           8103  88,002-012   88,002-012      88,002-009        88,002-009
**xxxx069E NEXT DBKEY DOES NOT POINT BACK TO NEXT RECORD IN SET
xxxx038I RECORDS DETECTED WALKING NEXT  FROM     88,002-012 . . .
xxxx039I NEXT  DBKEY POINTS TO PREVIOUSLY AUDITED RECORD IN SET
```

Figure 4.12: Set Errors -- Before Image Report

4.4.4 Set Errors -- After Image Report

Here is a description of the fields that make up the After Image Report. See Figure 4.13.

Lines 1 thru 5 - After Image Report header--this example shows a Simulated After Image report banner. This report shows what the database will look like if the fixes are performed by CA-IDMS/DB Audit.

Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,USERSETS.

Line 6 - Informative message--indicates the set for which a fix was simulated.

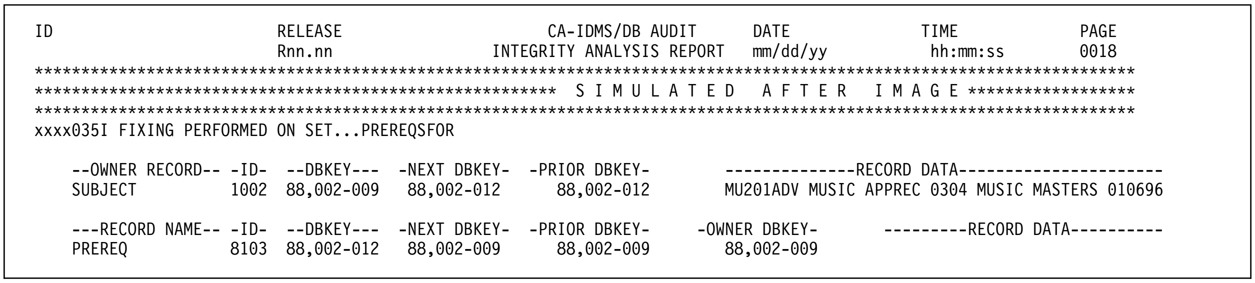
Line 8 - OWNER record information:

- **OWNER RECORD** - Name of the owner.
- **ID** - Record identifier.

- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior pointer (if present).
- **RECORD DATA** - First 59 characters of data.

Line 11 - Member record information:

- **RECORD NAME** - Name of the record.
- **ID** - Record identifier.
- **DBKEY** - Db-key of the database record.
- **NEXT DBKEY** - Next pointer.
- **PRIOR DBKEY** - Prior db-key (if present).
- **OWNER DBKEY** - Owner pointer (if present).
- **RECORD DATA** - First 45 characters of data.



```
ID                       RELEASE                     CA-IDMS/DB AUDIT      DATE              TIME             PAGE
                         Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss         0018
********************************************************************************************************************
*******************************************************  S I M U L A T E D   A F T E R   I M A G E *****************
********************************************************************************************************************
xxxx035I FIXING PERFORMED ON SET...PREREQSFOR

    --OWNER RECORD-- -ID-  --DBKEY---  -NEXT DBKEY-  -PRIOR DBKEY-        --------------RECORD DATA----------------------
    SUBJECT          1002  88,002-009   88,002-012      88,002-012        MU201ADV MUSIC APPREC 0304 MUSIC MASTERS 010696

    ---RECORD NAME-- -ID-  --DBKEY---  -NEXT DBKEY-  -PRIOR DBKEY-     -OWNER DBKEY-       ---------RECORD DATA----------
    PREREQ           8103  88,002-012   88,002-009      88,002-009        88,002-009
```

Figure 4.13: Set Errors -- After Image Report

4.4.5 Set Disconnections

An example of a record that was fully disconnected from a set is shown in Figures 4.14a and 4.14b. Records displayed on this section of the report are grouped by set name, and are members of either a mandatory automatic or a DISCONNECT set.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

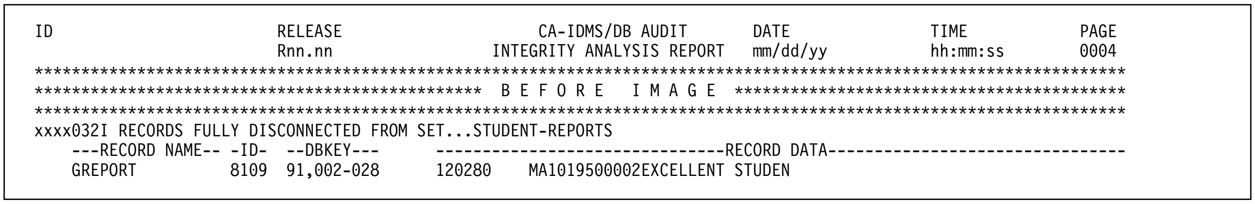
Line 6 - Informative message--indicates the set name where record was disconnected.

RECORD NAME - Name of the record.

ID - Record identifier.

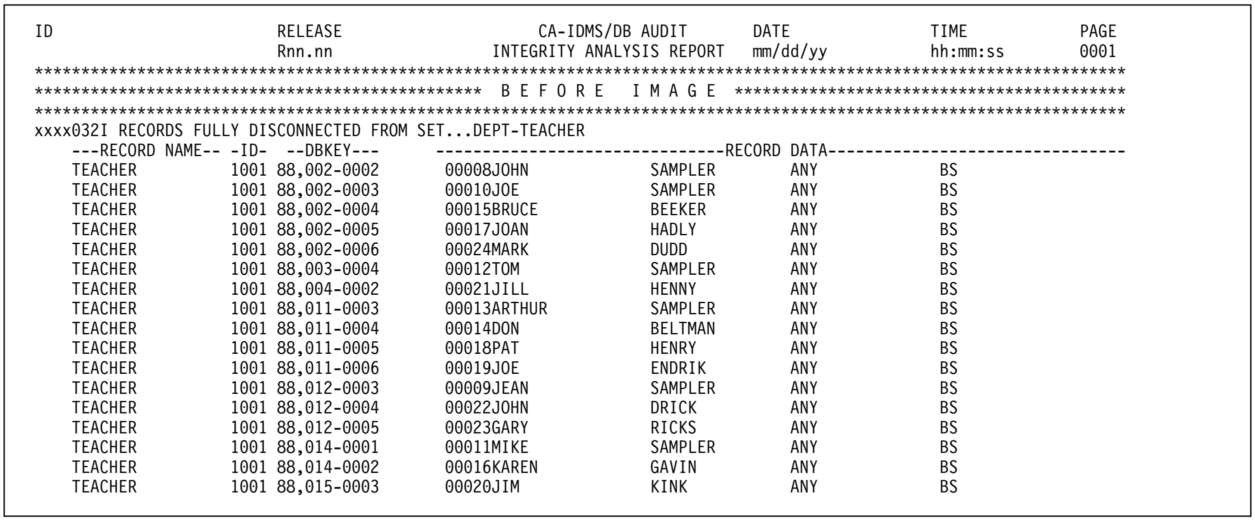
DBKEY - Db-key of the database record.

RECORD DATA - First 88 characters of data.



```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0004
*********************************************************************************************************************
************************************************  B E F O R E   I M A G E  ******************************************
*********************************************************************************************************************
xxxx032I RECORDS FULLY DISCONNECTED FROM SET...STUDENT-REPORTS
    ---RECORD NAME-- -ID-  --DBKEY---      -------------------------------RECORD DATA--------------------------------
    GREPORT          8109  91,002-028      120280    MA1019500002EXCELLENT STUDEN
```

Figure 4.14a: Set Disconnections



```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0001
*********************************************************************************************************************
************************************************  B E F O R E   I M A G E  ******************************************
*********************************************************************************************************************
xxxx032I RECORDS FULLY DISCONNECTED FROM SET...DEPT-TEACHER
    ---RECORD NAME-- -ID-  --DBKEY---      -------------------------------RECORD DATA--------------------------------
    TEACHER          1001 88,002-0002       00008JOHN             SAMPLER        ANY             BS
    TEACHER          1001 88,002-0003       00010JOE              SAMPLER        ANY             BS
    TEACHER          1001 88,002-0004       00015BRUCE            BEEKER         ANY             BS
    TEACHER          1001 88,002-0005       00017JOAN             HADLY          ANY             BS
    TEACHER          1001 88,002-0006       00024MARK             DUDD           ANY             BS
    TEACHER          1001 88,003-0004       00012TOM              SAMPLER        ANY             BS
    TEACHER          1001 88,004-0002       00021JILL             HENNY          ANY             BS
    TEACHER          1001 88,011-0003       00013ARTHUR           SAMPLER        ANY             BS
    TEACHER          1001 88,011-0004       00014DON              BELTMAN        ANY             BS
    TEACHER          1001 88,011-0005       00018PAT              HENRY          ANY             BS
    TEACHER          1001 88,011-0006       00019JOE              ENDRIK         ANY             BS
    TEACHER          1001 88,012-0003       00009JEAN             SAMPLER        ANY             BS
    TEACHER          1001 88,012-0004       00022JOHN             DRICK          ANY             BS
    TEACHER          1001 88,012-0005       00023GARY             RICKS          ANY             BS
    TEACHER          1001 88,014-0001       00011MIKE             SAMPLER        ANY             BS
    TEACHER          1001 88,014-0002       00016KAREN            GAVIN          ANY             BS
    TEACHER          1001 88,015-0003       00020JIM              KINK           ANY             BS
```

Figure 4.14b: Set Disconnections

4.4.6 Integrated Index Pointer Errors

An example of an integrated index set that contains pointer errors is shown in Figure 4.15.

CA-IDMS/DB Audit searches for this type of error if INDEX is specified in the AUDIT statement.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates the set found to contain pointer errors.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of database record.

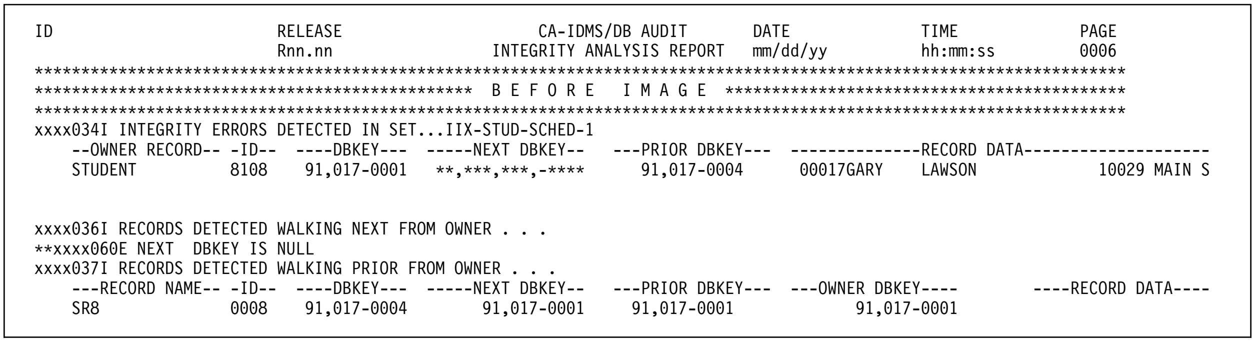
NEXT DBKEY - Next pointer.

PRIOR DBKEY - Prior pointer (if present).

OWNER DBKEY - Owner pointer (if present).

RECORD DATA - First portion of data.

Error message - Indicates the type of error detected.



```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE              TIME             PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy          hh:mm:ss         0006
*********************************************************************************************************************
***********************************************  B E F O R E   I M A G E  *******************************************
*********************************************************************************************************************
xxxx034I INTEGRITY ERRORS DETECTED IN SET...IIX-STUD-SCHED-1
    --OWNER RECORD-- -ID--  ----DBKEY---  -----NEXT DBKEY--   ---PRIOR DBKEY---  --------------RECORD DATA--------------------
    STUDENT          8108    91,017-0001   **,***,***,-****      91,017-0004      00017GARY    LAWSON             10029 MAIN S


xxxx036I RECORDS DETECTED WALKING NEXT FROM OWNER . . .
**xxxx060E NEXT  DBKEY IS NULL
xxxx037I RECORDS DETECTED WALKING PRIOR FROM OWNER . . .
    ---RECORD NAME-- -ID--  ----DBKEY---  -----NEXT DBKEY--   ---PRIOR DBKEY---  ---OWNER DBKEY----        ----RECORD DATA----
    SR8              0008    91,017-0004        91,017-0001     91,017-0001             91,017-0001
```

Figure 4.15: Integrated Index Pointer Errors

An example of a report that illustrates a set whose pointer errors have been fixed is shown in exhibit 4.16. CA-IDMS/DB Audit fixes this type of error if USERSETS is specified in the FIX statement. (Integrated index pointer errors were detected when INDEX was specified in the AUDIT statement.)

Here is a description of fields that make up the After Image Report.

Lines 1 thru 5 - After image header--this example shows a Simulated After Image report banner. This report shows what the database looks like if the fixes are performed by CA-IDMS/DB Audit.

Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,USERSETS.

Line 6 - Informative message--indicates the set that had a simulated fix performed.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

NEXT DBKEY - Next pointer.

PRIOR DBKEY - Prior pointer (if present).

OWNER DBKEY - Owner pointer (if present).

RECORD DATA - First several characters of data.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0008
*********************************************************************************************************************
**********************************************  S I M U L A T E D   A F T E R   I M A G E  **************************
*********************************************************************************************************************
xxxx035I FIXING PERFORMED ON SET...IIX-STUD-SCHED-1
    --OWNER RECORD-- -ID--   ----DBKEY----   ----NEXT DBKEY-   ----PRIOR DBKEY-     -------------------RECORD DATA------------
    STUDENT          8108    91,006-0001        91,006-0004     91,006-0004         00018DAVID     LAWSON         10029 MAIN S

    ---RECORD NAME-- -ID--   ----DBKEY----   ----NEXT DBKEY-   ----PRIOR DBKEY--    --OWNER DBKEY---          --RECORD DATA--
    SR8              0008    91,006-0004        91,006-0001     91,006-0001             81,006-0001
```

Figure 4.16: Integrated Index Pointer Errors

4.4.7 Integrated Index Orphans

The report in Figure 4.17 shows a list of records detected as orphans of integrated index sets. A record is an orphan record of an integrated index set if the upper level db-key does not point to the correct record in the level above it.

CA-IDMS/DB Audit searches for orphan records of integrated index sets if INDEX is specified in the AUDIT statement.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates that integrated index orphans were found.

SET NAME - Integrated index set name.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the record.

OWNER DBKEY - Owner pointer.

UPPER DBKEY - Db-key of the correct SR8 record in the level above.

MEM-UPPER-DBKEY - Upper level pointer of orphan record (member or lower level SR8)--the pointer in error.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0013
*********************************************************************************************************************
*************************************************  B E F O R E   I M A G E  *****************************************
*********************************************************************************************************************
xxxx059I INTEGRATED INDEX ORPHAN RECORDS DETECTED......


    ----SET NAME--   -RECORD NAME--    -ID-     ---DBKEY---   ---OWNER DBKEY--    ----UPPER DBKEY--  -MEM-UPPER-DBKEY-
    IIX-STUD-ACT-1    SCHEDULE         8107     91,010-0007        91,018-0001          91,018-0013        91,010-0001
    IIX-STUD-ACT-1    SCHEDULE         8107     91,010-0027        91,018-0001          91,018-0013        91,010-0001
    IIX-STUD-LNAME    SR8              0008     92,129-0005        92,129-0001          92,129-0029        92,129-0008
    IIX-STUD-LNAME    SR8              0008     92,129-0011        92,129-0001          92,129-0027        92,129-0007
    IIX-STUD-LNAME    SR8              0008     92,129-0012        92,129-0001          92,129-0024        92,129-0008
    IIX-STUD-LNAME    SR8              0008     92,129-0013        92,129-0001          92,129-0022        92,129-0003
    IIX-STUD-LNAME    SR8              0008     92,129-0017        92,129-0001          92,129-0032        92,129-0016
    IIX-STUD-LNAME    SR8              0008     92,129-0018        92,129-0001          92,129-0029        92,129-0008
    IIX-STUD-LNAME    SR8              0008     92,129-0019        92,129-0001          92,129-0024        92,129-0015
    IIX-STUD-LNAME    SR8              0008     92,129-0024        92,129-0001          92,129-0030        92,129-0017
    IIX-STUD-LNAME    SR8              0008     92,129-0025        92,129-0001          92,129-0036        92,129-0011
    IIX-STUD-LNAME    SR8              0008     92,129-0034        92,129-0001          92,129-0036        92,129-0026
    IIX-STUD-LNAME    STUDENT          8108     91,002-0001        92,129-0001          92,129-0012        92,129-0004
    IIX-STUD-LNAME    STUDENT          8108     91,002-0002        92,129-0001          92,129-0012        92,129-0005
    IIX-STUD-LNAME    STUDENT          8108     91,002-0003        92,129-0001          92,129-0019        92,129-0005
    IIX-STUD-LNAME    STUDENT          8108     91,002-0004        92,129-0001          92,129-0019        92,129-0014
    IIX-STUD-LNAME    STUDENT          8108     91,002-0005        92,129-0001          92,129-0038        92,129-0014
    IIX-STUD-LNAME    STUDENT          8108     91,002-0006        92,129-0001          92,129-0038        92,129-0023
    IIX-STUD-LNAME    STUDENT          8108     91,004-0001        92,129-0001          92,129-0014        92,129-0005
    IIX-STUD-LNAME    STUDENT          8108     91,007-0001        92,129-0001          92,129-0018        92,129-0004
    IIX-STUD-LNAME    STUDENT          8108     91,007-0002        92,129-0001          92,129-0018        92,129-0006
    IIX-STUD-LNAME    STUDENT          8108     91,007-0003        92,129-0001          92,129-0028        92,129-0006
    IIX-STUD-LNAME    STUDENT          8108     91,010-0001        92,129-0001          92,129-0025        92,129-0002
```

Figure 4.17: Integrated Index Orphans

4.4.8 Integrated Index Orphans Fixed

Here is a description of the fields that make up the After Image Report. See Figure 4.18.

Lines 1 thru 5 - After Image Report header--this example shows a Simulated After Image report banner. This report shows what the database will look like if the fixes are performed by CA-IDMS/DB Audit.

Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,INDEXORPHANS.

Line 6 - Informative message--indicates the set that had a simulated fix performed.

SET NAME - Integrated index set name.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

OWNER DBKEY - Owner pointer.

UPPER DBKEY - Db-key of the correct SR8 record in the level above.

MEM-UPPER-DBKEY - Upper level pointer of record (member or lower level SR8)--the pointer that was fixed.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0020
*********************************************************************************************************************
**********************************************  S I M U L A T E D   A F T E R   I M A G E  *************************
*********************************************************************************************************************
xxxx067I INTEGRATED INDEX ORPHANS FIXED.....


    ----SET NAME--  --RECORD NAME-     -ID-    ----DBKEY---    --OWNER DBKEY--   ---UPPER DBKEY----  -MEM-UPPER-DBKEY--
    IIX-STUD-ACT-1    SCHEDULE         8107     91,010-0007        91,018-0001          91,018-0013         91,018-0013
    IIX-STUD-ACT-1    SCHEDULE         8107     91,010-0027        91,018-0001          91,018-0013         91,018-0013
    IIX-STUD-LNAME    SR8              0008     92,129-0005        92,129-0001          92,129-0029         92,129-0029
    IIX-STUD-LNAME    SR8              0008     92,129-0011        92,129-0001          92,129-0027         92,129-0027
    IIX-STUD-LNAME    SR8              0008     92,129-0012        92,129-0001          92,129-0024         92,129-0024
    IIX-STUD-LNAME    SR8              0008     92,129-0013        92,129-0001          92,129-0022         92,129-0022
    IIX-STUD-LNAME    SR8              0008     92,129-0017        92,129-0001          92,129-0032         92,129-0032
    IIX-STUD-LNAME    SR8              0008     92,129-0018        92,129-0001          92,129-0029         92,129-0029
    IIX-STUD-LNAME    SR8              0008     92,129-0019        92,129-0001          92,129-0024         92,129-0024
    IIX-STUD-LNAME    SR8              0008     92,129-0024        92,129-0001          92,129-0030         92,129-0030
    IIX-STUD-LNAME    SR8              0008     92,129-0025        92,129-0001          92,129-0036         92,129-0036
    IIX-STUD-LNAME    SR8              0008     92,129-0034        92,129-0001          92,129-0036         92,129-0036
    IIX-STUD-LNAME    STUDENT          8108     91,002-0001        92,129-0001          92,129-0012         92,129-0012
    IIX-STUD-LNAME    STUDENT          8108     91,002-0002        92,129-0001          92,129-0012         92,129-0012
    IIX-STUD-LNAME    STUDENT          8108     91,002-0003        92,129-0001          92,129-0019         92,129-0019
    IIX-STUD-LNAME    STUDENT          8108     91,002-0004        92,129-0001          92,129-0019         92,129-0019
    IIX-STUD-LNAME    STUDENT          8108     91,002-0005        92,129-0001          92,129-0038         92,129-0038
    IIX-STUD-LNAME    STUDENT          8108     91,002-0006        92,129-0001          92,129-0038         92,129-0038
    IIX-STUD-LNAME    STUDENT          8108     91,004-0001        92,129-0001          92,129-0014         92,129-0014
    IIX-STUD-LNAME    STUDENT          8108     91,007-0001        92,129-0001          92,129-0018         92,129-0018
    IIX-STUD-LNAME    STUDENT          8108     91,007-0002        92,129-0001          92,129-0018         92,129-0018
    IIX-STUD-LNAME    STUDENT          8108     91,007-0003        92,129-0001          92,129-0028         92,129-0028
    IIX-STUD-LNAME    STUDENT          8108     91,010-0001        92,129-0001          92,129-0025         92,129-0025
```

Figure 4.18: Integrated Index Orphans Fixed

4.4.9 Additional Integrated Index Errors

Several examples of errors detected in integrated index sets are shown in Figures 4.19a, 4.19b, and 4.19c. These are errors other than orphan or pointer errors. CA-IDMS/DB Audit searches for these types of errors if you specify INDEX in the AUDIT statement. The specific type of error on each report is indicated by the error message.

Here is a description of the fields that make up the Before Image Reports.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates the set found to contain errors.

RECORD NAME - Name of the record.

ID - Record identifier.

Record not found error:

DBKEY - SR8 table entry db-key of a record not found.

UPPER DBKEY - Db-key of the SR8 record.

Record type incorrect error:

DBKEY - SR8 table entry db-key of a record of an incorrect type.

UPPER DBKEY - Db-key of SR8 record whose table entry points to the record that is an incorrect type for the set.

SR8 upper level pointer not HIGH-VALUES error:

DBKEY - Db-key of the record whose upper level pointer is incorrect.

UPPER DBKEY - Db-key of the owner of the SR8 record.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0005
*********************************************************************************************************************
************************************************ B E F O R E   I M A G E *******************************************
*********************************************************************************************************************
xxxx061I INDEX ENTRY DBKEY ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET IX-DEPT-NAME

    --RECORD NAME--  -ID-   ----DBKEY---    --UPPER DBKEY--
    DEPT             1000    88,008-0023        92,223-0002
**xxxx088E RECORD NOT FOUND - RECORD NAME IS DEPT
```

Figure 4.19a: Additional Integrated Index Errors

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0006
*********************************************************************************************************************
*********************************************** B E F O R E   I M A G E ********************************************
*********************************************************************************************************************
xxxx061I INDEX ENTRY DBKEY ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET IX-STUD-LNAME

    --RECORD NAME--  -ID-  -----DBKEY---    --UPPER DBKEY--
    CLASS            8106    89,003-0001        92,129-0025
**xxxx089E RECORD TYPE INCORRECT - RECORD NAME IS CLASS
```

Figure 4.19b: Additional Integrated Index Errors

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0008
*********************************************************************************************************************
************************************************ B E F O R E   I M A G E *******************************************
*********************************************************************************************************************
xxxx061I INDEX ENTRY DBKEY ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET STUDENT-SCHEDULE

    --RECORD NAME--  -ID-  -----DBKEY---   --UPPER DBKEY---
    SR8              0008    91,004-0008        91,003-0042
**xxxxx102E SR8 UPPER LEVEL POINTER IS NOT HIGH VALUES...
```

Figure 4.19c: Additional Integrated Index Errors

Here is a description of the fields that make up the Before Image Reports shown in Figures 4.20a and 4.20b.

Symbolic key errors:

DBKEY - Db-key of the record named.

SYMBOLIC KEY - Symbolic key of the record named.

UPPER DBKEY - Db-key of the SR8 record whose table entries were examined.

UPPER SYMB KEY - Symbolic key in the SR8 table entry. This symbolic key does not match the symbolic key in the SYMBOLIC KEY field.

Sort key sequence errors:

DBKEY - Db-key of SR8 record.

SORT KEY - Sort key entry in the SR8 record

UPPER DBKEY - Db-key of the upper level SR8 record whose db-key table entry points to the SR8 record identified in DBKEY field.

PREV SORT KEY - Sort key in the table entry in the upper level record identified in the UPPER DBKEY field. The sequence error means that this sort key does not match the sort key in the SORT KEY field.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0009
*********************************************************************************************************************
************************************************  B E F O R E   I M A G E  ******************************************
*********************************************************************************************************************
xxxx062I SYMBOLIC KEY ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED


    --RECORD NAME--  -ID-   -----DBKEY--   --SYMBOLIC KEY- --UPPER DBKEY---     --UPPER SYMB KEY---
    TEACHER          1001    88,003-0002   JONES                88,013-0017                   WHITE
    TEACHER          1001    88,014-0002   GAVIC                88,013-0003                   GOBEL
    TEACHER          1001    88,015-0004   GOBEL                88,013-0003                   GAVIC
```

Figure 4.20a: Additional Integrated Index Errors

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0010
*********************************************************************************************************************
************************************************  B E F O R E   I M A G E  ******************************************
*********************************************************************************************************************
xxxx063I SORT KEY SEQUENCE ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET IX-STUD-LNAME
xxxx064I SORT SEQUENCE ORDER IS DESCENDING
    ---RECORD NAME-- -ID- -----DBKEY----    ----SORT KEY---   -UPPER DBKEY-       ---PREV SORT KEY-
    SR8              0008    92,129-0025        91,010-0036     92,129-0030            89,003-0048
```

Figure 4.20b: Additional Integrated Index Errors

Here is a description of the fields that make up the Before Image Reports shown in Figures 4.20c and 4.20d.

Sort key sequence error--set sorted by symbolic key:

DBKEY - Db-key of the SR8 record.

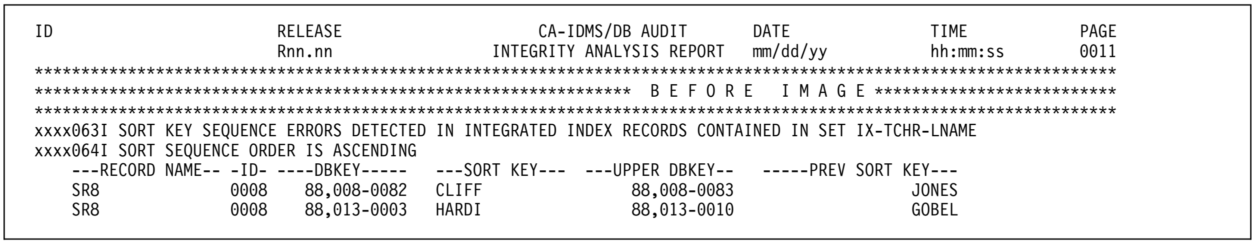
SORT KEY - Sort key entry in the SR8 record.

UPPER DBKEY - Db-key of the upper level SR8 record whose db-key table entry points to the SR8 record identified in item 9.

PREV SORT KEY - Sort key in the table entry in the upper level record identified in the UPPER DBKEY field. The sequence error means that this sort key does not match the sort key in SORT KEY field.

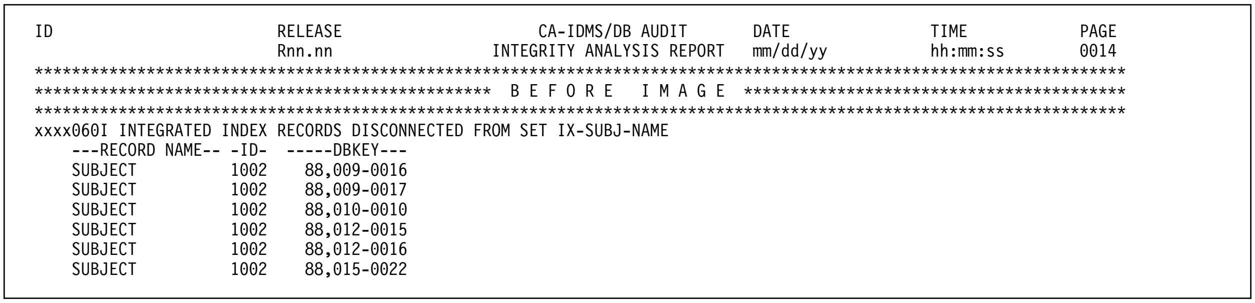
Disconnected records:

List of records disconnected from the indicated integrated index set.



```
ID                        RELEASE                    CA-IDMS/DB AUDIT       DATE              TIME            PAGE
                          Rnn.nn               INTEGRITY ANALYSIS REPORT    mm/dd/yy          hh:mm:ss         0011
*********************************************************************************************************************
***************************************************************  B E F O R E   I M A G E ***************************
*********************************************************************************************************************
xxxx063I SORT KEY SEQUENCE ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET IX-TCHR-LNAME
xxxx064I SORT SEQUENCE ORDER IS ASCENDING
    ---RECORD NAME-- -ID- ----DBKEY-----   ---SORT KEY---  ---UPPER DBKEY--   -----PREV SORT KEY---
    SR8              0008    88,008-0082   CLIFF                88,008-0083                   JONES
    SR8              0008    88,013-0003   HARDI                88,013-0010                   GOBEL
```

Figure 4.20c: Additional Integrated Index Errors



```
ID                        RELEASE                    CA-IDMS/DB AUDIT       DATE              TIME            PAGE
                          Rnn.nn               INTEGRITY ANALYSIS REPORT    mm/dd/yy          hh:mm:ss         0014
**********************************************************************************************************************
*************************************************  B E F O R E   I M A G E  *****************************************
**********************************************************************************************************************
xxxx060I INTEGRATED INDEX RECORDS DISCONNECTED FROM SET IX-SUBJ-NAME
    ---RECORD NAME-- -ID-  -----DBKEY---
    SUBJECT          1002    88,009-0016
    SUBJECT          1002    88,009-0017
    SUBJECT          1002    88,010-0010
    SUBJECT          1002    88,012-0015
    SUBJECT          1002    88,012-0016
    SUBJECT          1002    88,015-0022
```

Figure 4.20d: Additional Integrated Index Errors

4.4.10 From/Thru Page Range Errors

An example of a record that falls outside its designated from/thru page range is shown in Figure 4.21. The record does not participate in a set that was audited and that does not contain other integrity errors.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates a from/thru page range error.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

FROM PAGE - From page number.

THRU PAGE - Thru page number.

RECORD DATA - First 67 characters of data.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0005
*********************************************************************************************************************
************************************************  B E F O R E   I M A G E  ******************************************
*********************************************************************************************************************
xxxx031I RECORD DBKEYS OUTSIDE SPECIFIED FROM/THRU PAGE RANGES . . .
    ---RECORD NAME-- -ID- --DBKEY---    FROM PAGE     THRU PAGE   ----------------------RECORD DATA----------------------
    TEACHER          1001 88,003-004       88,002        88,002   00012TOM       SAMPLER    ANY     BS           01040000
    TEACHER          1001 88,003-006       88,002        88,002   00034KEVIN     SINGER     ART     BS           01059400
    TEACHER          1001 88,003-005       88,002        88,002   00026RONNIE    RHYME      MUSIC   BS           01055000
    TEACHER          1001 88,004-003       88,002        88,002   00029CLARENCE  FORD       MATH    BS,MS,PHD    01058000
    TEACHER          1001 88,006-002       88,002        88,002   00030JOHN      SOSA       ART     BA           01059000
    TEACHER          1001 88,004-002       88,002        88,002   00021JILL      ENDRIC     ANY     BS           01049000
    TEACHER          1001 88,015-005       88,002        88,002   00031MAUREEN   HARRIS     SCIENCE BS           01059100
    TEACHER          1001 88,005-002       88,002        88,002   00033ROGER     CABE       MUSIC   NOT ENOUGH   01059300
    TEACHER          1001 88,006-003       88,002        88,002   00035TRACIE    LERNER     MUSIC   TOO MUCH     01059500
    TEACHER          1001 88,003-003       88,002        88,002   00003THOMAS    BLACK      MATH    BS/MATH      00950000
    TEACHER          1001 88,012-001       88,002        88,002   00004JACK      CORD       MATH    BS/MATH      00960000
    TEACHER          1001 88,015-002       88,002        88,002   00007DANNY     DUDLEY     MATH    BS/MATH      00990000
    TEACHER          1001 88,011-001       88,002        88,002   00013ARTHUR    AMBER      ANY     BS           01041000
    TEACHER          1001 88,011-003       88,002        88,002   00018PAT       DRIER      ANY     BS           01046000
    TEACHER          1001 88,011-002       88,002        88,002   00014DON       SELTER     ANY     BS           01042000
    TEACHER          1001 88,011-004       88,002        88,002   00019JOE       HENNY      ANY     BS           01047000
    TEACHER          1001 88,012-003       88,002        88,002   00009JEAN      ABLE       ANY     BS           01010000
    TEACHER          1001 88,012-006       88,002        88,002   00027BURT      NOLL       SCIENCE NONE         01056000
    TEACHER          1001 88,012-004       88,002        88,002   00022JOHN      HENRY      ANY     BS           01049100
    TEACHER          1001 88,012-005       88,002        88,002   00023GARY      WEBB       ANY     BS           01049200
    TEACHER          1001 88,012-007       88,002        88,002   00028VINCENT   RIDER      MUSIC   BS           01057000
    TEACHER          1001 88,014-001       88,002        88,002   00011MIKE      MOLER      ANY     BS           01030000
    TEACHER          1001 88,014-002       88,002        88,002   00016KAREN     GAVIN      ANY     BS           01044000
```

Figure 4.21: From/Thru Page Range Errors

4.4.11 Orphans of Sets Other Than Integrated Index Sets

An example of a record that could not be associated to any OWNER record in a user set is shown in Figure 4.22. Records can fall into this category only if WALKNEXT was requested.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates that records were detected as orphans.

SET NAME - Set name where record is an orphan.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

NEXT DBKEY - Next pointer.

PRIOR DBKEY - Prior pointer (if present).

OWNER DBKEY - Owner pointer (if present).

RECORD DATA - First 28 characters of data.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0011
*********************************************************************************************************************
*************************************************  B E F O R E   I M A G E  *****************************************
*********************************************************************************************************************
xxxx043I ORPHANS NOT CONNECTED TO ANY OWNER RECORD . . .


    ----SET NAME----  --RECORD NAME---ID-   --DBKEY---NEXT DBKEY- -PRIOR DBKEY-  -OWNER DBKEY-  -----RECORD DATA------
    CLASS-SCHEDULE    SCHEDULE      8107   89,008-052  89,016-022    89,016-021     89,017-004  FAM102 D 0030000000005
    CLASS-SCHEDULE    SCHEDULE      8107   89,016-021  89,008-052    89,016-022     89,017-004  FAM102 D 0020000000005
    CLASS-SCHEDULE    SCHEDULE      8107   89,016-022  89,016-021    89,008-052     89,017-004  FAM102 D 0040000000005
```

Figure 4.22: Orphans of Sets Other Than Integrated Index Sets

4.4.12 Fragmented Record Errors

An example of a fragmented (variable length) record that could not be properly assembled is shown in Figure 4.23. The record does not participate in a set that was audited and that does not contain other integrity errors.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

Line 6 - Informative message--indicates a fragmented record assembly error.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

RECORD DATA - First 88 characters of the CONTROL portion of the data.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0003
*********************************************************************************************************************
*************************************************  B E F O R E   I M A G E  *****************************************
*********************************************************************************************************************
xxxx048I FRAGMENTED RECORDS NOT PROPERLY ASSEMBLED . . .
    ---RECORD NAME-- -ID-  ---DBKEY--     -----------------------CONTROL PORTION OF RECORD DATA----------------------
    DREPORT          8110  91,002-031
```

Figure 4.23: Fragmented Record Errors

4.4.13 LDEL Records Physically Deleted

An example where a logically deleted (LDEL) record was physically deleted from two sets is shown in Figure 4.24. The record was also erased since it was physically deleted from all sets where it participated as a member.

Here is a description of the fields that make up the After Image Report.

Lines 1 thru 5 - After Image Report header--this example shows a Simulated After Image Report banner. This report shows what the database will look like if the fixes are performed by CA-IDMS/DB Audit.

Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,LDEL.

Line 6 - Informative message--indicates physical deletion of LDEL records.

SET NAME - Indicates set name where record was physically deleted.

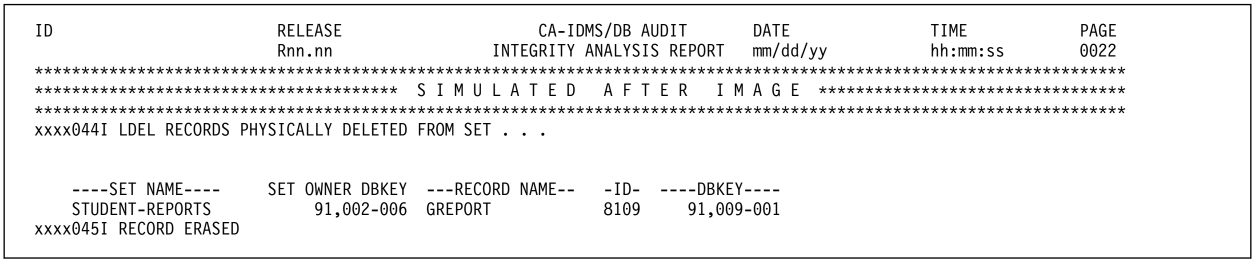
SET OWNER DBKEY - Db-key of the owner of the set where the record was deleted.

RECORD NAME - Name of the record.

ID - Record identifier.

DBKEY - Db-key of the database record.

Informative message - Indicates record was erased.



```
ID                        RELEASE                    CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0022
**********************************************************************************************************************
**************************************** S I M U L A T E D   A F T E R   I M A G E ***********************************
**********************************************************************************************************************
xxxx044I LDEL RECORDS PHYSICALLY DELETED FROM SET . . .


    ----SET NAME----     SET OWNER DBKEY  ---RECORD NAME--   -ID-  ----DBKEY----
    STUDENT-REPORTS          91,002-006  GREPORT            8109     91,009-001
xxxx045I RECORD ERASED
```

Figure 4.24: LDEL Records Physically Deleted

4.4.14 Page Errors

Two examples of the Before Image Report fields for Page Errors illustrating invalid Header/Footer page numbers are shown in Figures 4.25a and 4.25b. Page errors are grouped by area name.

Note: The RECDATA operand does not affect the format of Page Error displays.

Here is a description of the fields that make up the Before Image Report.

Lines 1 thru 5 - Before Image Report header.

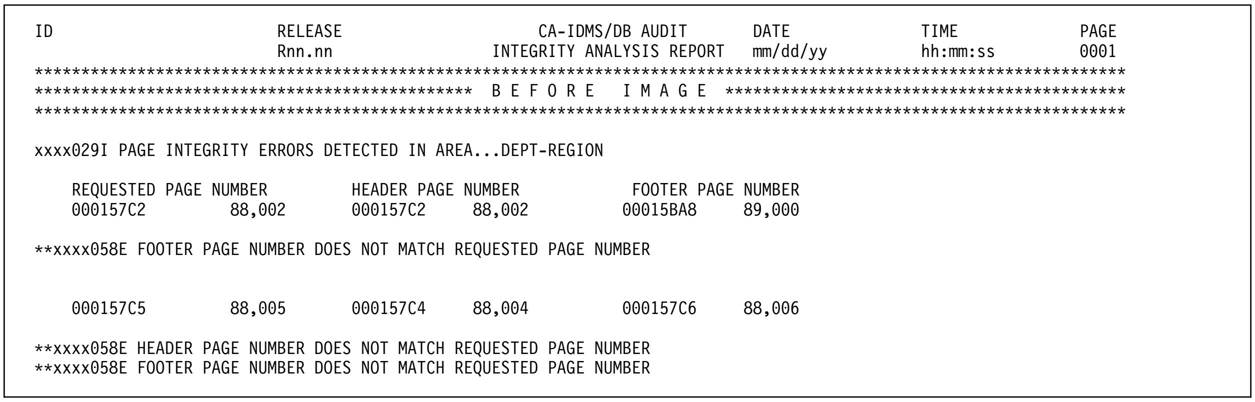
Line 6 - Informative message--indicates area where page errors were detected.

REQUESTED PAGE NUMBER - Requested page number in hexadecimal and requested page number in decimal.

HEADER PAGE NUMBER - Header page number in hexadecimal and header page number in decimal.

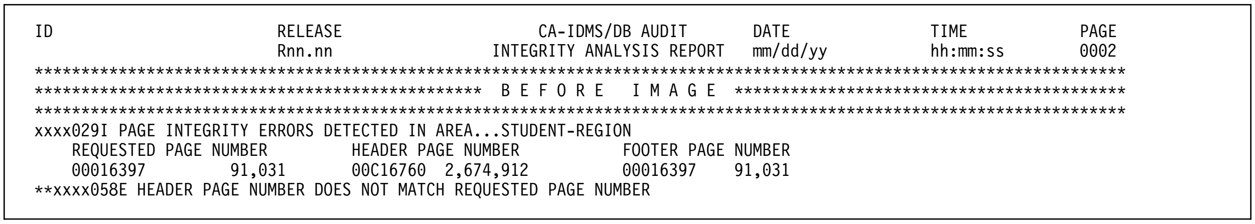
FOOTER PAGE NUMBER - Footer page number in hexadecimal and footer page number in decimal.

Error message - Indicates the type of error detected.



```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE              TIME             PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy          hh:mm:ss         0001
*********************************************************************************************************************
***********************************************  B E F O R E   I M A G E  *******************************************
*********************************************************************************************************************

xxxx029I PAGE INTEGRITY ERRORS DETECTED IN AREA...DEPT-REGION

    REQUESTED PAGE NUMBER         HEADER PAGE NUMBER            FOOTER PAGE NUMBER
    000157C2         88,002       000157C2     88,002          00015BA8     89,000

**xxxx058E FOOTER PAGE NUMBER DOES NOT MATCH REQUESTED PAGE NUMBER


    000157C5         88,005       000157C4     88,004          000157C6     88,006

**xxxx058E HEADER PAGE NUMBER DOES NOT MATCH REQUESTED PAGE NUMBER
**xxxx058E FOOTER PAGE NUMBER DOES NOT MATCH REQUESTED PAGE NUMBER
```

Figure 4.25a: Page Errors



```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn                 INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0002
*********************************************************************************************************************
************************************************  B E F O R E   I M A G E  ******************************************
*********************************************************************************************************************
xxxx029I PAGE INTEGRITY ERRORS DETECTED IN AREA...STUDENT-REGION
    REQUESTED PAGE NUMBER         HEADER PAGE NUMBER           FOOTER PAGE NUMBER
    00016397         91,031       00C16760  2,674,912          00016397    91,031
**xxxx058E HEADER PAGE NUMBER DOES NOT MATCH REQUESTED PAGE NUMBER
```

Figure 4.25b: Page Errors

Here is a description of the fields that make up the After Image Report. See Figures 4.26a and 4.26b.

Lines 1 thru 5 - After Image Report header--this example shows a Simulated After Image Report banner. This report shows what the database will look like if the fixes are performed by CA-IDMS/DB Audit.

Note: The fixes shown on this report have not been performed. In order for the fixes to be performed, submit another job that specifies FIX=UPDATE,PAGES.

Line 6 - Informative message--indicates area where pages were fixed.

REQUESTED PAGE NUMBER - Requested page number in hexadecimal and requested page number in decimal.

HEADER PAGE NUMBER - Header page number in hexadecimal and header page number in decimal.

FOOTER PAGE NUMBER - Footer page number in hexadecimal and footer page number in decimal.

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn               INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0016
*****************************************************************************************************************
***************************************  S I M U L A T E D   A F T E R   I M A G E **********************************
*****************************************************************************************************************
xxxx030I FIXING PERFORMED ON PAGES IN AREA...DEPT-REGION
    REQUESTED PAGE NUMBER         HEADER PAGE NUMBER           FOOTER PAGE NUMBER
    000157C2         88,002       000157C2     88,002          000157C2    88,002
    000157C5         88,005       000157C5     88,005          000157C5    88,005
```

Figure 4.26a: Page Errors

```
ID                        RELEASE                     CA-IDMS/DB AUDIT       DATE               TIME            PAGE
                          Rnn.nn               INTEGRITY ANALYSIS REPORT   mm/dd/yy           hh:mm:ss        0017
*****************************************************************************************************************
***************************************  S I M U L A T E D   A F T E R   I M A G E **********************************
*****************************************************************************************************************
xxxx030I FIXING PERFORMED ON PAGES IN AREA...STUDENT-REGION
    REQUESTED PAGE NUMBER         HEADER PAGE NUMBER           FOOTER PAGE NUMBER
    00016397         91,031       00016397     91,031          00016397    91,031
```

Figure 4.26b: Page Errors

Chapter 5. Operations

This chapter describes auditing methods and operational procedures. It provides information for determining storage requirements and for operating CA-IDMS/DB Audit efficiently. It also provides sample JCL for executing CA-IDMS/DB Audit in an OS/390, VSE/ESA , or VM/ESA environment.

5.1 Database Access

CA-IDMS/DB Audit does not use CA-IDMS to access a user database. Instead, CA-IDMS/DB Audit uses an access method that was developed to provide for CA-IDMS/DB Audit's unique needs for accessing a CA-IDMS database.

All CA-IDMS conventions regarding locking the database against access are observed. While CA-IDMS/DB Audit is executing, you must ensure that no central version (CV) or local CA-IDMS update jobs access the database.

5.2 Functions

CA-IDMS/DB Audit performs three functions:

- **AUDITING** - searching a database for physical integrity errors
- **FIXING** - using information collected during auditing to correct specified types of physical integrity errors
- **REPORTING** - displaying the physical integrity errors found during auditing and displaying the corrections if the errors were fixed

Corrections can be either simulated or real. You can select simulated fixing and examine the corrections on the reports before actually updating the database.

5.3 Modes of Operation

CA-IDMS/DB Audit performs auditing, fixing, and reporting functions. You can execute CA-IDMS/DB Audit to perform any one or combination of these functions:

- **Mode 1** - AUDIT only
- **Mode 2** - AUDIT and FIX (Simulated or Real)
- **Mode 3** - FIX only (Simulated or Real)
- **Mode 4** - REPORTS only

Modes 1, 2, and 3 produce an Integrity Analysis Report unless you specifically suppress it. Mode 4 always produces an Integrity Analysis Report.

All modes automatically produce an Audit Report; you cannot suppress it.

5.3.1 Using the Extract File

The AUDIT function of CA-IDMS/DB Audit generates an extract file. This extract file is used during the FIX and the REPORTS phases. You can run FIX and REPORTS either in the same execution as AUDIT, or in a later execution.

The recommended procedure is to run CA-IDMS/DB Audit twice: first in Mode 2 using simulated FIX, and then in Mode 3 using update FIX. This procedure allows you to review the results of the audit with proposed corrections before updating the database. It is important not to change the database between the audit and fix jobs and to retain the extract file.

In an OS/390 environment, you can catalog the extract file at the end of the audit and simulated fix job. In all environments, do not delete the extract file or write over it before running the fix phase.

5.4 Inclusive and Exclusive Processing

CA-IDMS/DB Audit enables you to selectively include or exclude subschema areas, records, or sets from an audit run. You can also mix inclusive and exclusive processing. For example, you can include named areas and exclude records or sets from those areas. You can also include a record and exclude particular sets that the record owns.

The following parameter options influence inclusive and exclusive processing. These parameters are fully explained in the “Parameters” chapter.

```
PROCESS Statement: STARTWITH

AREA Statement:    NOACCESS/SWEEP/NOSWEEP
                   BEGIN and END

RECORD Statement:  NOACCESS/SWEEP/NOSWEEP

SET Statement:     AUDIT/NOAUDIT
```

To understand inclusive and exclusive processing, you must understand the relationship between STARTWITH and the AREA, RECORD, and SET parameter options.

STARTWITH defines only the starting point of how much is to be audited based on the subschema named in the PROCESS statement. AREA, RECORD, and SET options further identify the scope of processing to be performed.

To initiate **inclusive processing**, first specify STARTWITH=NONE to exclude all portions of the database from processing. Then supply AREA, RECORD, and SET statements to select portions of the database to be included in processing.

To initiate **exclusive processing**, first specify STARTWITH=ALL to include all portions of the database for processing. Then supply AREA, RECORD, and SET statements to select portions of the database to be excluded from processing. Because STARTWITH=ALL is the default parameter, it need not be explicitly specified. For clarity, this guide always specifies STARTWITH=ALL for exclusive processing.

Mixed processing combines elements of inclusive and exclusive processing to make very select auditing possible. For example, you could include an area and exclude sets from within that area. You can specify either STARTWITH=ALL or STARTWITH=NONE depending on which portions of the database are to be audited.

5.5 Processing Examples

This section contains examples of parameter syntax used to initiate various kinds of inclusive, exclusive, and mixed audit processing. The processing examples presented on the following pages are based on the subschema TESTSUB, presented as a data structure diagram in the "Concepts" chapter of this guide.

5.5.1 Parameter Guidelines

General guidelines on parameter syntax structure and audit processing follow:

- The STARTWITH option and AREA, RECORD, and SET statements are hierarchical: STARTWITH has the broadest impact, and SET has the narrowest. You can use a lower level parameter to reverse an effect caused by a higher level parameter, thereby mixing inclusive and exclusive processing.
- AREA options propagate to logically related records and sets.
- RECORD options propagate to logically related areas and sets.
- SET options propagate to logically related areas and records.
- Use the positive subparameters (AUDIT and SWEEP) in AREA, RECORD, and SET statements to include elements that were excluded through STARTWITH=NONE.
- Specify negative subparameters (NOACCESS and NOAUDIT) to exclude elements that were included through STARTWITH=ALL.
- NOSWEEP is unique in that it can be considered as either a positive (allows access) or negative (do not sweep) option, depending upon use.
- If you specify NOACCESS for either AREA or RECORD statements, do not specify AUDIT for SET statements that are associated with the same area or record.

  Similarly, if you specify NOACCESS on an AREA statement, do not specify a record SWEEP or set AUDIT for records and sets that are associated with the area.
- CA-IDMS/DB Audit identifies conflicting parameters as either errors or warnings. Processing terminates if any parameter presents a direct conflict between AREA, RECORD, or SET parameters. Audit processing continues when conflicting options produce warnings, but some options may be overridden, especially when conflicts arise during propagation.
- The Audit Report Areas/Records/Sets report section documents the processing that will occur.
- In the PROCESS statement, specify the SCAN option to allow you to review the Areas/Records/Sets section of the Audit Report and verify what processing is to take place before CA-IDMS/DB Audit actually begins that processing.

5.5.2 Inclusive Processing Examples

The following examples are based on the subschema TESTSUB and invoke inclusive processing.

Example 1 - Audit one set only

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=NONE
AUDIT,USERSETS
SET=ROOM-CLASS,AUDIT
```

When you audit a single set, you do not have to specify the areas because CA-IDMS/DB Audit automatically accesses the areas containing the set. The area LOC-REGION contains the set owner and is automatically set to SWEEP.

If you specify Standard as the audit type, the member area CLASS-REGION is made accessible. If QuickCheck is the audit type, member area CLASS-REGION is set to SWEEP.

Example 2 - Sweep a single area

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=NONE
AUDIT,USERSETS
AREA=DEPT-REGION,SWEEP
```

In this example, only those sets whose owners reside in DEPT-REGION are audited. Specifically, the following sets are audited:

```
DEPT-TEACHER
DEPT-ACADEMIC
DEPT-GENERAL
PREREQSFOR
PREREQSARE
SUBJECT-CLASS
TEACHER-CLASS
```

The index (IX) sets would not be audited because their owners are in the INDEX-REGION.

Example 3 - Sweep more than one area

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=NONE
AUDIT,USERSETS
AREA=DEPT-REGION,SWEEP
AREA=INDEX-REGION,SWEEP
AREA=CLASS-REGION,SWEEP
AREA=STUDENT-REGION, SWEEP
SET=PERIOD-CLASS,AUDIT
```

All index and user sets except ROOM-CLASS are audited. In this particular case, however, it would be easier to use exclusive processing (STARTWITH=ALL) to exclude the one set ROOM-CLASS.

Example 4 - Audit all sets owned by a single record

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=NONE
AUDIT,USERSETS
RECORD=SUBJECT,SWEEP
```

In this case, the following sets are audited:

```
PREREQSFOR
PREREQSARE
SUBJECT-CLASS
```

To sweep a record, you do not need to specify SWEEP for the area where the record resides because CA-IDMS/DB Audit handles that automatically. All sets that the record owns are swept when you specify record sweep.

Under QuickCheck, sweeping an owned set implies sweeping the area containing the records that participate in the set. Multiple areas may also be swept if the record owns cross-area sets. A **cross-area set** is a set whose owners and members reside in different areas. Under Standard auditing, sweeping all owned sets may also cause multiple areas to be accessed if the record owns cross-area sets.

Example 5 - Confine processing to specific pages

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=NONE,AUDITTYPE=STANDARD
AUDIT,USERSETS
AREA=STUDENT-REGION,BEGIN=00010,END=00030
```

Because BEGIN and END imply SWEEP, you do not need to specify SWEEP.

In this example, the audit type can only be Standard. You cannot specify USERSETS with QuickCheck if you also want to perform partial page range audits with BEGIN and END. For QuickCheck, the USERSETS option requires entire area page ranges to be swept in order to detect errors properly. Therefore, you cannot specify BEGIN or END in QuickCheck when USERSETS is also specified.

The same restriction against specifying BEGIN or END applies to QuickCheck CALCSETS as well.

5.5.3 Exclusive Processing Examples

The examples contained in this section represent parameter syntax for exclusive processing. For clarity, the default STARTWITH=ALL is included in all of the following examples.

Example 1 - Sweep and audit all areas except CLASS-REGION

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=ALL
AUDIT,USERSETS
AREA=CLASS-REGION,NOACCESS
```

Any set whose owner or members reside in CLASS-REGION is not audited. Specifically, the following sets are not audited:

```
SUBJECT-CLASS
TEACHER-CLASS
ROOM-CLASS
PERIOD-CLASS
CLASS-SCHEDULE
STUDENT-SCHEDULE
```

Example 2 - Do not sweep area CLASS-REGION, but access its member records from cross-area sets

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=ALL,AUDITTYPE=STANDARD
AUDIT,USERSETS
AREA=CLASS-REGION,NOSWEEP
```

A cross-area set is a set whose owners and members reside in different areas. In this example, all sets whose owners reside outside of CLASS-REGION are audited, even if the members reside in CLASS-REGION. Consequently, the only set not audited is CLASS-SCHEDULE.

To satisfy the conditions of Example 2, the audit type must be Standard. Example 3, which follows, demonstrates the effect of specifying QuickCheck as the audit type with all other parameters being the same.

Example 3 - Do not audit owned sets in area CLASS-REGION with QuickCheck

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=ALL,AUDITTYPE=QUICKCHECK
AUDIT,USERSETS
AREA=CLASS-REGION,NOSWEEP
```

Again, in this example, only the CLASS-SCHEDULE set is not audited. Because the audit type is QuickCheck, however, the sets with owner records that point into the NOSWEEP CLASS-REGION from another area must be set either to AUDIT or NOAUDIT. This parameter is required because, by definition, NOSWEEP does not impact member sets, yet QuickCheck must sweep member records if their owners are swept and their sets are audited. One way to audit owned sets that point into CLASS-REGION is to audit one of the sets that point into the CLASS-REGION, thereby forcing the CLASS-REGION to be swept:

```
AREA=CLASS-REGION, NOSWEEP
SET=SUBJECT-CLASS, AUDIT
```

Example 4 - Audit an entire subschema except for sets IX-SUBJ-NAME and IX-STUD-LNAME

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=ALL
AUDIT,USERSETS
SET=IX-SUBJ-NAME,NOAUDIT
SET=IX-STUD-LNAME,NOAUDIT
```

Integrated index sets that are stored in the same area page range must be audited--or not audited--together. If IX-SUBJ-NAME and IX-TCHR-LNAME were in the same page range, you would have no choice but to set them both to AUDIT or to NOAUDIT.

5.5.4 Mixed Processing Examples

Mixed processing combines elements of inclusive and exclusive processing. The following examples demonstrate a few ways mixed processing can be used.

Example 1 - Sweep a single area, but do not audit the set TEACHER-CLASS contained in area DEPT-REGION:

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=NONE
AUDIT,USERSETS
AREA=DEPT-REGION,SWEEP
SET=TEACHER-CLASS,NOAUDIT
```

In this example, the following sets are audited:

```
DEPT-TEACHER
DEPT-ACADEMIC
DEPT-GENERAL
PREREQSFOR
PREREQSARE
SUBJECT-CLASS
```

If QuickCheck is the audit type, then CLASS-REGION is additionally swept for CLASS records because the set SUBJECT-CLASS is audited.

Example 2 - Sweep and audit all areas, but, within the area DEPT-REGION, audit only the set DEPT-TEACHER

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=ALL
AUDIT,USERSETS
AREA=DEPT-REGION,NOSWEEP
SET=DEPT-TEACHER,AUDIT
```

Although the AREA statement specifies NOSWEEP for the area DEPT-REGION, it must be swept in order to audit set DEPT-TEACHER. Other sets in DEPT-REGION are not audited because the audit set overrides the area NOSWEEP for records in DEPT-TEACHER only. However, the area NOSWEEP accomplishes the NOAUDIT for the other sets whose owners are in DEPT-REGION.

Example 3 - Specify NOACCESS in DEPT-REGION and AUDIT DEPT-TEACHER in DEPT-REGION

```
PROCESS,SUBSCHEMA=TESTSUB,STARTWITH=ALL
AUDIT,USERSETS
AREA=DEPT-REGION,NOACCESS
SET=DEPT-TEACHER,AUDIT
```

The parameters in this example create a more serious conflict than those in Example 2. In this case, CA-IDMS/DB Audit cannot reasonably determine whether area NOACCESS or set AUDIT is more important to the user. Consequently, processing cannot continue and CA-IDMS/DB Audit displays an error message.

5.6 Operating Preparations

Before running CA-IDMS/DB Audit, prepare for processing as follows:

- **Prevent additional updates** - During the time the database is being audited, CA-IDMS/DB Audit follows the normal CA-IDMS conventions regarding locking the area against access. If you specify READYMODE=RETRIEVAL, you must ensure that no other update jobs run against an area being audited.

  If you run CA-IDMS/DB Audit twice (once for audit and once for fix), you must prevent any CA-IDMS updates from taking place in the audited areas until after you apply the fixes. If any updates are made between the time the audit is run and the time the fixes are applied, the corrections may not be valid or new integrity errors may be introduced. Similarly, do not perform any updates until you have completed the REPORT function.

- **Back up the database** - CA-IDMS/DB Audit does not write before and after images of changed data to a journal file. Therefore, you should back up the database before initiating fix processing and again after fix processing.

Perform the following preparations within CA-IDMS/DB Audit:

- **Include index area in the subschema** - If you specify AUDIT INDEX, you must copy the index areas and associated records and sets into the subschema.

  If you specify INDEX, copy the area containing the index members and owners into the subschema.

- **In the subschema, define all members of sets to be audited** - You cannot successfully audit a set unless you define all the set's member record types in the subschema. This includes all members of the CALC set residing in areas for which CALCSET processing will occur.

- **Specify correct SUBSCHEMA/DMCL and load/core image library** - Be sure to specify the correct subschema for the database you wish to audit. If you specify the wrong subschema, CA-IDMS/DB Audit can incorrectly flag a large number of pages as having header or footer errors. Because CA-IDMS/DB Audit does not access the subschema/DMCL from a dictionary, the correct copies must be in the job's step library or core image library.

- **Specify RETRIEVAL only for runs with no updates** - In the PROCESS statement, specify RETRIEVAL as the readymode for runs that do not update the database. Legitimate runs are AUDIT, FIX=SIMULATE or AUDIT, REPORTS.

  With READYMODE=RETRIEVAL, CA-IDMS/DB Audit does not lock the database, so you must ensure that no other job updates the database. Otherwise, if the database being audited is in the process of being updated, CA-IDMS/DB Audit may incorrectly indicate that the database contains integrity errors.

- **Specify UPDATE for actual fixing** - When you are ready to fix the database, specify READYMODE=UPDATE in the job that is to perform the fixing. Be sure that the database to be fixed is not modified between the AUDIT and the FIX or REPORTS jobs.

- **Allow sufficient time to audit the database** - Because CA-IDMS/DB Audit may access every record and set occurrence in the database several times, it can take a long time to audit the database. Recognize that in most cases Standard processing takes longer than QuickCheck processing.
- **Specify DISK if sets to be audited are large** - Large sets, including integrated index sets, require large amounts of storage for the tables used in processing. If you specify DISK to contain these tables, you must allocate two VSAM files before executing CA-IDMS/DB Audit.
- **Allocate enough space for the DBKYWORK file** - This step applies to Standard auditing only. The DBKEYWORK file can require a substantial amount of space if you audit integrated indexes whose areas contain many mandatory-automatic (MA) index set members. For every MA index set member, CA-IDMS/DB Audit writes two 30-byte records to the DBKEYWORK file. If a member record belongs to more than one index set, two records are written for each set to which it belongs.
- **Allocate enough space for INDXEXTR and INDXWORK files** - The INDXEXTR and INDXWORK files are used for integrated index orphan extract records and for other integrated index error extract records.

  For each index orphan found, CA-IDMS/DB Audit writes one index orphan extract record to the INDXEXTR file. If you are fixing (simulate or update) index orphans, CA-IDMS/DB Audit also writes one index orphan extract record to the INDXWORK file. Be sure to allocate enough sort work space for integrated index orphans.
- **Include a sufficient area portion** - CA-IDMS/DB Audit allows partial area auditing and fixing. CA-IDMS/DB Audit does not allow fixing, however, when you perform a partial area audit and either of the following situations exists:
  - A set containing an integrity error points into a second set with errors, and the owner of the second set is outside the partial area being swept.
  - Records are detected as orphans of a set containing integrity errors, and the owner of the set is outside the partial area being swept.
- **Review the output** - Review the reports CA-IDMS/DB Audit generates to understand the errors detected and the way they have been fixed.

Validate the logical integrity of any set occurrences CA-IDMS/DB Audit fixed. Because CA-IDMS/DB Audit cannot analyze logical integrity conditions enforced by user programs, you must correct any logical errors once the physical errors have been fixed. You can use the CA-IDMS utility IDMSBCF, FIX PAGE function to fix errors that CA-IDMS/DB Audit does not fix. For a list of the types of integrity errors that CA-IDMS/DB Audit does not fix, refer to Chapter 2, “Concepts.”

5.7 Performance Factors

Several factors affect the performance of CA-IDMS/DB Audit:

- **Size of the database** - The number of pages, the number of record occurrences, and the number of set types affect the amount of time required to audit the database
- **Record location mode** - The more member records that are stored VIA the audited set and the less VIA overflow count, the better the performance
- **Auditing method** - QuickCheck auditing is faster than Standard auditing because it requires less database access.

5.7.1 Estimating USERSETS I/Os with Standard Auditing

You can use the following formula to calculate the approximate number of I/Os to audit USERSETS in your database (when you have not specified WALKNEXT or DISCONNECT). Additional I/Os are required if CA-IDMS/DB Audit detects integrity errors.

I/Os = p + n + (s * v)

p – represents the number of pages swept

n – represents the number of CALC or DIRECT member record occurrences for sets being audited

s – the number of set occurrences whose VIA members are stored in a different page range than the set owner

v – represents the number of VIA member record types stored in different page ranges than the set owner

Refer to Figure 5.1 for an example. If you were to audit USERSETS only in AREA-1 (AREA-2 is NOSWEEP), you would estimate the number of I/Os as follows:

```
   10,000 pages swept
 + 75,000 type E records for set F-E
 + 120,000 type G records for set E-G
 + 120,000 type G records for set H-G
 + 185,000 type D-B set occurrences
 = 510,000 I/Os
```

(**v**=1, because all B records are in AREA-2.)

If record type G were stored VIA set E-G, it would take approximately 120,000 fewer I/Os to audit the database for a total of 390,000 I/Os. If, in addition, record type E were stored VIA set F-E, it would take approximately 75,000 fewer I/Os to audit the database for a total of 315,000 I/Os.

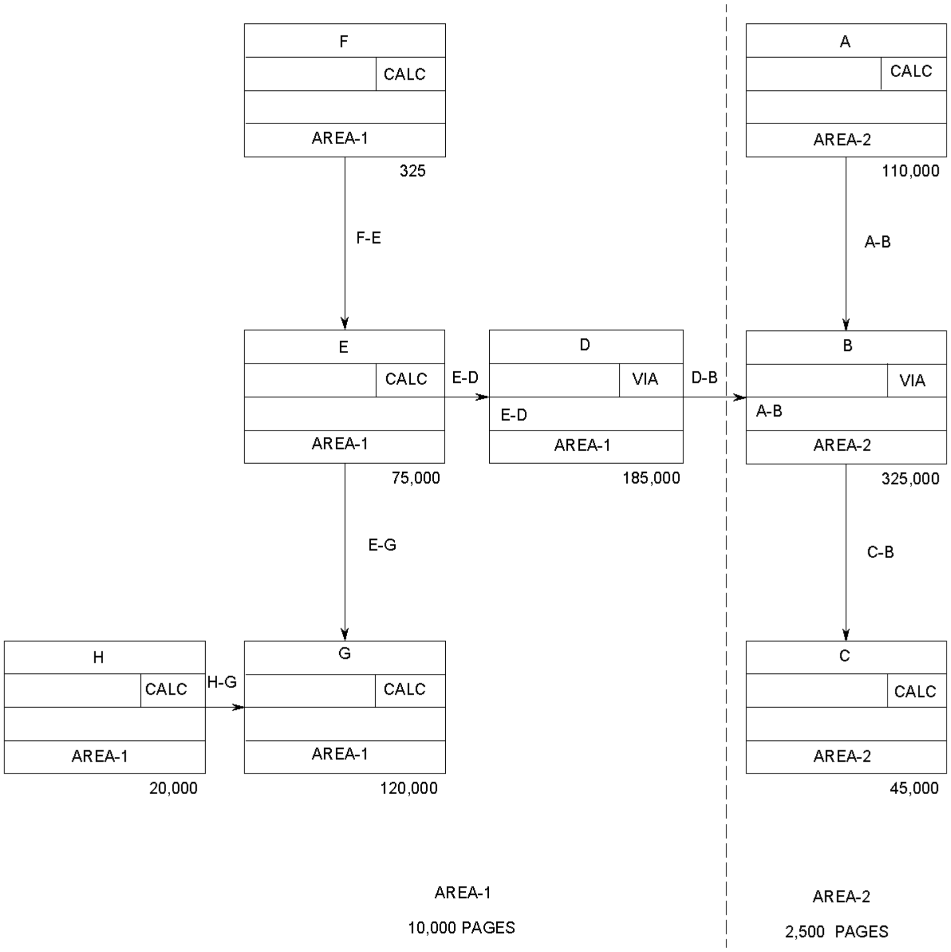


Figure 5.1: AREA-1 and AREA-2, Estimating USERSETS I/Os

5.7.2 Estimating USERSETS I/Os with QuickCheck Auditing

The number of I/Os required to audit USERSETS during QuickCheck processing is approximately equal to the number of pages swept.

5.7.3 Estimating CALCSETS I/Os with Standard or QuickCheck Auditing

The approximate number of I/Os to audit only CALCSETS should be equal to the number of pages swept, because members of the CALC set are VIA the SR1 record for the page to which the record calculated.

5.7.4 Estimating WALKNEXT I/Os with Standard Auditing

WALKNEXT applies to Standard auditing only; it does not apply to QuickCheck.

WALKNEXT causes each record in the area to be accessed several times. When you specify WALKNEXT, each set occurrence is walked as many times as there are member record occurrences plus once more for the owner. The result is a significant increase in processing time over normal set walk processing. WALKNEXT not only requires numerous I/Os to complete the audit, but also exponentially increases the number of calls to DBIO regardless of whether or not members are stored VIA the sets being audited.

Use the following formula to calculate the approximate number of I/Os required to audit your database with WALKNEXT. Additional I/Os are needed if CA-IDMS/DB Audit detects integrity errors.

```
I/Os = p + (o *l2)
```

p — represents the number of pages swept

o — represents the number of owner records for each CALC or DIRECT set

l — represents the average set length of the CALC or DIRECT set

Figure 5.2 provides an example. If you are auditing USERSETS within AREA-1 with WALKNEXT (set H-G is the only non-VIA set), you would estimate the number of I/Os as follows:

```
(l = 120,000/20,000 = 6)

   10,000 pages swept
 + 720,000 I/Os for set H-G (20,000 owner records * 36)
 = 730,000 I/Os
```

If the same area were audited without WALKNEXT, the number of I/Os would be approximately:

```
   10,000 pages swept
 + 120,000 type G records for set H-G
 = 130,000 I/Os
```

Use the WALKNEXT option only if you know or suspect that there are record occurrences that would not be found during normal set walking. To determine whether your database contains orphan records, refer to 4.4.5, "Set Disconnections."

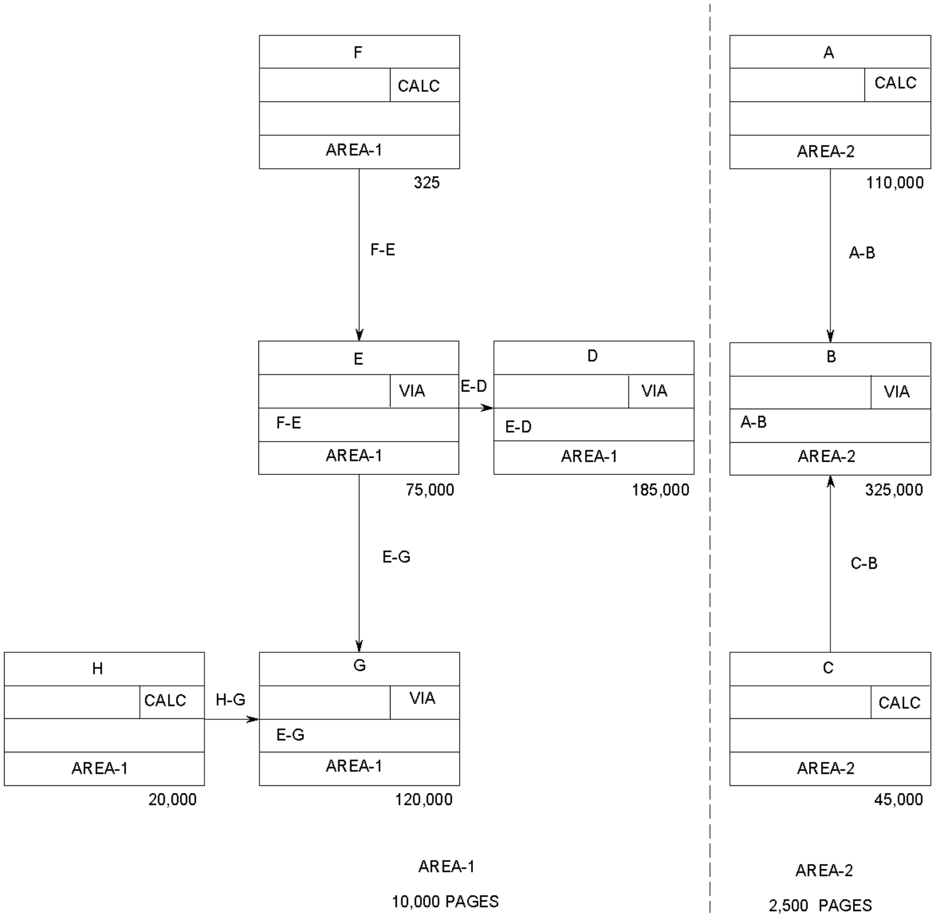


Figure 5.2: AREA-1 and AREA-2, Estimating WALKNEXT I/Os

5.7.5 Estimating DISCONNECT I/Os with Standard Auditing

The DISCONNECT option slows performance because of the large number of records retrieved. CA-IDMS/DB Audit examines each record retrieved in an area sweep to see if it is a member of a set that is to be audited for disconnections. If all of such a record's db-keys for the set being audited are null (equal to high-values), CA-IDMS/DB Audit indicates that the record is fully disconnected from the set. If a next, prior, or owner pointer is not null, CA-IDMS/DB Audit retrieves the record indicated by the pointer to determine whether it is a valid owner or member of the set being audited.

DISCONNECT not only requires numerous I/Os to complete the audit, but also exponentially increases the number of calls to DBIO regardless of whether or not the sets being audited are VIA or non-VIA (DIRECT or CALC).

You can calculate the approximate number of I/Os required to audit your database for set disconnections (assuming no integrity errors are detected and most of the records are not fully disconnected) using the following formula:

```
I/Os = p + (m * k)
```

p — represents the number of pages swept

m — represents the number of member records in the area being swept

k — represents the number of non-VIA member record db-keys in the record's prefix

For example, in Figure 5.3, auditing for set disconnections in AREA-2 would take approximately:

```
    2,500 pages swept
+ 975,000 I/Os for set C-B (325,000 * 3)
 = 977,500 I/Os
```

Disconnect processing is independent of CALCSETS and USERSETS auditing. The number of I/Os generated by disconnect processing is in addition to the number of I/Os generated by CALCSETS and USERSETS auditing (except for the number of I/Os generated by area sweeping).

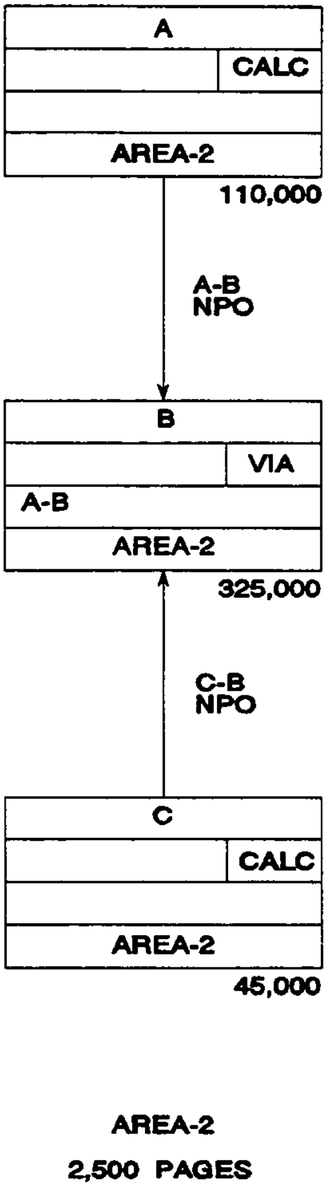


Figure 5.3: AREA-2, Estimating DISCONNECT I/Os

5.7.5.1 Effect of Specifying Both WALKNEXT and DISCONNECT

When you specify both WALKNEXT and DISCONNECT, walknext processing takes precedence over disconnect processing. During area sweeping, if CA-IDMS/DB Audit retrieves a record that is a member of a set being audited for disconnections, it examines the next and prior db-keys. If either the next or the prior db-key is not null, CA-IDMS/DB Audit performs walknext processing rather than simply retrieving the records pointed to by the next, prior, and owner db-keys.

5.7.6 Improving Performance

CA-IDMS/DB Audit provides several options for improving processing performance:

- Omit one or more areas if the database has several areas.
- Perform a partial area audit. Partial area audits always require less processing time than full area audits, although you must decide what action to take for any sets that cross area boundaries. You can perform a partial area audit in one of the following ways:
  - Select specific set types to audit by including or excluding other areas, records, and sets.
  - Specify BEGIN and END pages in an AREA statement if the database has only one large area (Standard auditing only). You can fix errors in the partial area if the owners of all sets involved are included in the partial area. See 5.6, “Operating Preparations” for more information.
- Perform a QuickCheck audit rather than a Standard audit.

5.8 Storage Requirements

You need to consider CA-IDMS/DB Audit's storage requirements for both main storage and external file storage.

5.8.1 Main Storage

CA-IDMS/DB Audit storage requirements vary depending on the size of the subschema table being processed, the maximum database page size, the number of buffers, sort storage requirements, whether or not VSAM is used, and the selected AUDIT options.

After processing a valid PROCESS statement, CA-IDMS/DB Audit loads the subschema and DMCL load modules or phases into main storage and builds a pseudo-subschema (PSUB) from the real CA-IDMS subschema. Once the pseudo-subschema is built, the real CA-IDMS subschema and DMCL are deleted from main storage. After parameter processing is completed, a BIND RUN-UNIT is issued. At this time, all of DBIO's work areas are obtained.

CA-IDMS/DB Audit then calculates the amount of storage required for its work areas and issues a GETMAIN/GETVIS to obtain this amount from the region/partition. The amount of main storage not obtained by CA-IDMS/DB Audit (and DBIO) is available for operating system use (such as CA-IDMS/DB Audit's file buffers, VSAM, and sort).

Note: In a VSE/ESA environment, all of CA-IDMS/DB Audit's and DBIO's work areas are obtained from the GETVIS area, not from the Problem Program Area (PPA). Only the SORT program and associated sort work areas are allocated in the PPA.

Use the formula shown in Figure 5.4 to estimate the amount of virtual storage required to run CA-IDMS/DB Audit.

VIRTUAL STORAGE =	
CA-IDMS/DB Audit SIZE	330K. Size of CA-IDMS/DB Audit module.
+ CA-IDMS/DB Audit Buffers	Sum of block size times the number of buffers for each non-database file.
+ PSUB	Approximately 3 times the total size of the subschema and DMCL load modules or core image phases.
+ Database Buffers	Maximum page size times number of buffers specified in the DMCL named in the subschema.
+ Sort Storage	Use your installation's default sort size.
+ VSAM Storage	Optional. Used if database has VSAM files or the DISK option is used. Refer to the appropriate system manual to determine the size of the ACB, EXT list, and RPL. If the DISK option is used to store the db-key table and the db-key save table on VSAM disk, add two 4K buffers.
+ DBKEYTBL	SETLIMIT value times 9.6 (result is in bytes). Used only if the CORE option is used and auditing CALCSETS and/or USERSETS and/or INDEX. If a SETLIMIT value is not specified, use 28K.
+ DBKEY Save Table	SETLIMIT value times 8 (result is in bytes). Used only if the CORE option is used and auditing CALCSETS and/or USERSETS and/or INDEX. If a SETLIMIT value is not specified, use 24K.
+ Record Bind Area	Total length of all audited records defined in the subschema (including prefixes) plus an additional 64 bytes for each record being audited. If auditing integrated index sets, add the largest page size of containing SR8 records.
+ Sort Key Area	Total length of the symbolic keys of all records with sets being audited. This area is the work area required for validating a member record's symbolic key within a sorted set.
+ Page Bind Area	Used for page audit. Sum of number of areas audited times 40.
+ IIX Level Save Area	If auditing integrated index sets, add the maximum SR8 size times the maximum number of index levels.

Figure 5.4: Formula for Estimating Virtual Storage

5.8.2 External File Storage

CA-IDMS/DB Audit makes use of five sequential files: EXTRACTS, WORKFILE, DBKYWORK, INDXEXTR, and SORT. The following sections explain how to estimate space allocation for each file.

Allocate space only for the files you need for a particular CA-IDMS/DB Audit run. You must always allocate space for EXTRACTS and WORKFILE. If you are processing index sets, allocate space for INDXEXTR and INDXWORK. If you are processing mandatory automatic index sets, allocate space for DBKYWORK.

5.8.2.1 EXTRACTS File

Space allocation for the EXTRACTS file depends on the number of errors detected during the audit phase and the size of the extract record. Determine the size of the extract record using this formula:

```
Record Size = 260 + length of longest symbolic key +
     4 bytes for the record descriptor word (RDW)
```

Because it is difficult to estimate the number of errors, you can allocate the EXTRACTS file to tape. In a typical situation, a DASD cylinder allocation of 10,2 should be adequate.

5.8.2.2 WORKFILE File

Space allocation for the WORKFILE file should be twice the space allocation of the EXTRACTS file. In a typical situation, a DASD cylinder allocation of 20,4 should be adequate.

5.8.2.3 DBKYWORK File

To estimate the total number of blocks for the DBKYWORK file, use the following formula:

```
Record size = 30

Total Number of Blocks = (30 * (total # integrated index member records of MA sets)) * 2
                                                   block size
```

In a typical situation, a DASD cylinder allocation of 5,2 should be adequate.

5.8.2.4 INDXEXTR and INDXWORK Files

Space for the INDXEXTR and INDXWORK files is primarily used for integrated index orphan records and, secondarily, for integrated index errors.

If your database contains a small number of member records in integrated index sets, then you can estimate the total blocks using this formula:

```
Total Number of Blocks=((total # integrated index member records)*record size)*0.5
                                                      block size

 Record Size = 76 + length of member symbolic key +
               length of owner symbolic key +
               4 bytes for the record descriptor word (RDW)
```

If your database contains a large number of member records in integrated index sets, multiply the result by 3. In a typical situation, a DASD cylinder allocation of 20,4 should be adequate.

Both INDXEXTR and INDXWORK should use the same space allocation.

5.8.2.5 SORT File

For standard auditing, space allocation for the SORT file depends on the number of error extract records created in the audit phase. If your database has numerous errors or the sets containing errors are long, the extract file may also be large. In that case, you may need more than four sort work datasets.

If you are auditing integrated index sets and you have index orphans, then you need enough sort space to accommodate twice the number of index orphans written to the index extract file.

In most cases, the number of extracts created is small. In a typical situation with Standard auditing, four sort work files with a cylinder allocation of 5,2 should be adequate. The total sort work file allocation should be twice the space allocation of the EXTRACTS file.

For QuickCheck auditing, allocate space to sort 26-byte records for every record swept in the database. For each record swept, CA-IDMS/DB Audit creates a 26-byte extract record and a 26-byte match record for each set in which the swept record participates. Additionally, two 26-byte records are created for each SR7 and SR8 for index sets in the swept area.

```
 Total Number of Bytes to Sort = 26 * 2mn
```

m — represents the number of records swept

n — represents the number of sets of which those records are participants

2*mn* — represents the number of records sorted

26 — represents the length of sort record

5.9 CA-IDMS/DB Audit Sample JCL and Key

Sample JCL and key for running CA-IDMS/DB Audit in an OS/390, VSE/ESA, or VM/ESA environment is listed below and supplied in source library member USAEXEC. The version of USAEXEC you receive is specific to your operating environment.

Sample IDCAMS control statements for allocating work files are contained in source library member USADEFN. This member is the same for all environments.

5.9.1 Allocating Work Files for the DISK Option

If you specify DISK in the AUDIT statement, you must allocate two direct access disk files: one for the db-key hash table and one for the db-key save table. These direct access files may be VSAM or, for OS/390 and VM/ESA only, BDAM. If these files are allocated as VSAM, they must be defined through IDCAMS. This optional step is identical for all environments.

Sample control statements for VSAM file allocation are shown in Figure 5.5 and are contained in source library member USADEFN. A key to the variables (shown in **bold**) is shown in Figure 5.6.

```
DELETE (vsam.dbkhash) CLUSTER
DELETE (vsam.dbksave) CLUSTER
DEFINE CLUSTER (-
            NAME(vsam.dbkhash) -
            RECORDS(record.numbersh) -
            NUMBERED -
            RECORDSIZE(4084 4084) -
            SPEED -
            usetype -
            FILE(dname) -
            VOLUMES(volume.name) -
            ) -
      DATA  ( -
            NAME(vsam.dbkhash.DATA) -
            ) -
DEFINE CLUSTER ( -
            NAME(vsam.dbksave) -
            RECORDS(record.numberss) -
            NUMBERED -
            RECORDSIZE(4064 4064) -
            SPEED -
            usetype -
            VOLUMES(volume.name) -
            ) -
      DATA  ( -
            NAME(vsam.dbksave.DATA) -
            )
```

Figure 5.5: Control Statements for Allocating VSAM Work Files for the DISK Option

vsam.dbkhash — The dataset name of the VSAM file for the db-key table. This file is needed only if you specify DISK in the AUDIT statement.

record.number — The number of records to be allocated in the VSAM work space for the db-key table. The number of records is equal to:

$$\frac{1.2 * \text{SETLIMIT-value}}{510} + 1$$

For example, if the SETLIMIT value is 3000 (default), you should allocate:

```
((1.2 * 3000/510)   + 1 = 8
```

vsam.dbksave — The dataset name of the VSAM file for the db-key save table. This file is needed if you specify DISK in the AUDIT statement.

record.numberss — The number of records to be allocated in the VSAM work space for the db-key save table. The number of records depends on the number of next, prior, or owner pointer errors that you expect CA-IDMS/DB Audit to find while set walking. Up to 127 pointer errors can be accommodated by one record. Therefore, you should allocate 1 primary record with extra secondary space, unless you expect more errors. An allocation of (1 10) should be adequate. The maximum number of records is equal to:

$$\frac{\text{SETLIMIT-value}}{127} + 1$$

usetype — The appropriate type: REUSE or UNIQUE. If you specify UNIQUE, you must DELETE and DEFINE this cluster prior to each execution of CA-IDMS/DB Audit.

volume.name — Volume to contain the cluster or component.

dname — The DD name of the JCL statement defining the file. Depending upon your environment, the parameter is optional.

Figure 5.6: Key to Figure 5.5

5.10 OS/390 Operations

Sample JCL for using CA-IDMS/DB Audit in an OS/390 environment is presented below in three parts:

1. (Optional) Allocating work files for the DISK option.
2. Allocating CA-IDMS/DB Audit files.
3. Running CA-IDMS/DB Audit.

An explanation of various restart options, showing the use of files created in a previous audit run, follows the JCL.

5.10.1 OS/390 JCL

5.10.1.1 1. Allocating Work Files for the DISK Option

(USADEFN) If you specify DISK in the AUDIT statement and decide to use VSAM for the direct access files, you must allocate two VSAM files to disk: one for the db-key hash table and one for the db-key save table. See the 5.9.1, “Allocating Work Files for the DISK Option” and Figure 5.5 above for sample IDCAMS control statements for file allocation.

5.10.1.2 2. Allocating Files

The sample OS/390 JCL shown below in Figures 5.7a and 5.7b deletes any previous files and then allocates space for the general files and index files. Target or Distribution source library member USAEXEC contains similar sample JCL. The key to the variables (shown in **bold**) is shown in Figures 5.8a and 5.8b.

```
//*          OPTIONAL: SAMPLE JCL TO DELETE
//*          PREVIOUSLY CREATED FILES
//*
//DELETE  EXEC  PGM=IEFBR14
//EXTRACTS  DD  DSN=dbaudit.extract,
//              DISP=(MOD,DELETE)'
//              UNIT=disk,
//              SPACE=(TRK,0)
//WORKFILE  DD  DSN=dbaudit.workfile,
//              DISP=(MOD,DELETE),
//              UNIT=disk,
//              SPACE=(TRK,0)
//DBKYWORK  DD  DSN=dbaudit.dbkywork,
//              DISP=(MOD,DELETE),
//              UNIT=disk,
//              SPACE=(TRK,0)
//INDXEXTR  DD  DSN=dbaudit.indxextr
//              DISP=(MOD,DELETE),
//              UNIT=disk,
//              SPACE=(TRK,0)
//INDXWORK  DD  DSN=dbaudit.indxwork,
//              DISP=(MOD,DELETE),
//              UNIT=disk,
//              SPACE=(TRK,0)
//*         ALLOCATING GENERAL FILES
//*
//ALLOCATE EXEC  PGM=IEFBR14
//EXTRACTS  DD  DSN=dbaudit.extract,
//              DISP=(,CATLG,DELETE),
//              VOL=SER=volser,
//              UNIT=disk,
//              SPACE=(CYL,(pe,se)),
//              DCB=BLKSIZE=blkse
//WORKFILE  DD  DSN=dbaudit.workfile,
//              DISP=(,CATLG,DELETE),
//              VOL=SER=volser,
//              UNIT=disk,
//              SPACE=(CYL,(pw,sw)),
//              DCB=BLKSIZE=blkse
```

Figure 5.7a: OS/390 JCL for Allocating Files

```
//*      ALLOCATING INDEX FILES
//*
//DBKYWORK  DD  DSN=dbaudit.dbkywork,
//              DISP=(,CATLG,DELETE),
//              VOL=SER=volser,
//              UNIT=disk,
//              SPACE=(CYL,(pd,sd)),
//              DCB=BLKSIZE=blksd
//INDXEXTR  DD  DSN=dbaudit.indxextr,
//              DISP=(,CATLG,DELETE),
//              VOL=SER=volser,
//              UNIT=disk,
//              SPACE=(CYL,(pi,si)),
//              DCB=BLKSIZE=blksi
//INDXWORK  DD  DSN=dbaudit,indxwork,
//              DISP=(,CATLG,DELETE),
//              VOL=SER=volser,
//              UNIT=disk,
//              SPACE=(CYL,(pi,si)),
//              DCB=BLKSIZE=blksi
```

Figure 5.7b: OS/390 JCL for Allocating Files

```
//*        OPTIONAL: SAMPLE JCL TO DELETE & ALLOCATE
//*                  THE 2 DIRECT ACCESS FILES AS BDAM
//*
//DELETE2  EXEC PGM=IEFBR14
//DBKHASH  DD   DSN=dbaudit.dbkhash,
//              DISP=(MOD,DELETE),
//              UNIT=disk,
//              SPACE=(TRK,0),
//DBKSAVE  DD   DSN=dbaudit.dbksave,
//              DISP=(MOD,DELETE),
//              UNIT=disk,
//              SPACE=(TRK,0),
//ALLOCAT2 EXEC PGM=IEFBR14
//DBKHASH  DD   DSN=dbaudit.dbkhash,
//              DISP=(CATLG,DELETE),
//              VOL=SER=volser,
//              UNIT=disk,SPACE=(4084,(blkcnt.hash)),
//              DCB=DSORG=DA
//DBKSAVE  DD   DSN=dbaudit.dbksave,
//              DISP=(CATLG,DELETE),
//              VOL=SER=volser,
//              UNIT=disk,SPACE=(4064,(blkcnt.save)),
//              DCB=DSORG=DA
```

Figure 5.7c: OS/390 JCL for Allocating Files

5.10.1.3 Key to Sample OS/390 JCL for Allocating Files

dbaudit.extract — The dataset name assigned to your extract file.

disk — An appropriate unit designation for each file.

dbaudit.workfile — The dataset name assigned to your work file.

dbaudit.dbkywork — The dataset name assigned to your db-key work file. This file is needed only if you are auditing or fixing integrated index sets, and you have mandatory-automatic index sets.

dbaudit.indxextr — The dataset name assigned to your integrated index extract file. This file is needed only if you are auditing or fixing integrated index sets.

dbaudit.indxwork — The dataset name assigned to your integrated index work file. This file is needed only if you are auditing or fixing integrated index sets.

dbaudit.dbkhash — The dataset name assigned to your BDAM db-key hash table. This file is only needed if you specify the DISK option of the AUDIT statement.

dbaudit.dbksave — The dataset name assigned to your BDAM db-key save table. This file is only needed if you specify the DISK option of the AUDIT statement.

volser — The volume serial number of the disk that will contain a particular file.

pe,se — The primary/secondary space allocation for the EXTRACTS file allocated to DASD. Space allocation for a DASD file depends on the number of errors detected during the audit phase the maximum record size is 512 bytes and must be blocked by the user (as noted in **blkse**). Because it is difficult to estimate the number of errors, you can allocate this file to tape. For most cases, DASD can be used. For example, if you are using a 3330 device with a blocksize of 6144, one cylinder can hold 456 extract records. A primary/secondary specification of **10,2** should be adequate.

blkse — The block size of the EXTRACTS and WORKFILE files. The maximum record size is 512 bytes. A blocksize of 6144 is recommended for all device types.

pw,sw — The primary/secondary space allocation for the WORKFILE file allocated to DASD. Space allocation should be twice the space allocation of the EXTRACTS file.

A primary/secondary allocation of **20,4** should be adequate.

pd,sd — The primary/secondary space allocation for the DBKYWORK file allocated to DASD. To estimate the total number of blocks, multiply the total number of integrated index member records of mandatory-automatic sets by 30 (record length) times 2. The result is the total number of bytes. Divide by the block size to get the total number of blocks.

A primary/secondary allocation of **5,2** should be adequate.

blksd — The block size of the DBKWORK file. The record size is 30 bytes. A block size of 6000 is recommended for all device types.

pi,si — The primary/secondary space allocation for the INDXEXTR file and for the INDXWORK file, allocated to DASD. This space is primarily used for integrated index orphan records and secondarily for integrated index errors. If you have inserted a small number of member records in integrated index sets in the database, then you can estimate the total blocks by multiplying the total number of integrated index member records times the maximum record length (588) times 0.5, dividing by the block size. If a large number of member records have been inserted in integrated index sets, multiply the result by 3.

A primary/secondary allocation of **20,4** should be adequate.

blksi — The block size of the INDXEXTR and INDXWORK files. The maximum record size is 588 bytes. A blocksize of 6144 is recommended for all device types.

blkcnt.hash — The number of blocks for the BDAM db-key hash file. See Figure 5.6.

blkcnt.save — The number of blocks for the BDAM db-key save file. See Figure 5.6.

Figure 5.8: Key to Figure 5.7

5.10.1.4 3. Running CA-IDMS/DB Audit

Sample OS/390 JCL to audit and simulate fixing the database described by your subschema is shown in Figures 5.9a and 5.9b. A key to the variables (shown in **bold**) and file use is shown in Figures 5.10a and 5.10b. Similar JCL is contained in Target or Distribution source library member USAEXEC.

```
//*
//* THIS JCL IS FOR AUDIT AND SIMULATED FIXING
//*
//DBAUDIT EXEC  PGM=USADRVR,
//              REGION=#K,
//              PARM='NOSPIE/'
//STEPLIB   DD   DSN=your.loadlib,
//              DISP=SHR
//          DD   DSN=idms.loadlib,
//              DISP=SHR
//SORTLIB   DD   DSN=sort.loadlib,
//              DISP=SHR
//SORTWK01  DD   UNIT=disk,
//              SPACE=(CYL,(ps,ss))
//SORTWK02  DD   UNIT=disk,
//              SPACE=(CYL,(ps,ss))
//SORTWK03  DD   UNIT=disk,
//              SPACE=(CYL,(ps,ss))
//SORTWK04  DD   UNIT=disk,
//              SPACE=(CYL,(ps,ss))
//*
//dbfil01   DD   DSN=user.dbfil01,
//              DISP=SHR
//          .
//          .
//          .
//dbfilnn   DD   DSN=user.dbfilnn,
//              DISP=SHR
```

Figure 5.9a: OS/390 JCL to Run CA-IDMS/DB Audit

```
//*GENERAL FILES NEEDED FOR ALL RUNS.
//*
//EXTRACTS  DD   DSN=dbaudit.extract,
//               DISP=OLD
//WORKFILE  DD   DSN=dbaudit.workfile,
//               DISP=OLD
//*
//sortmsg   DD   SYSOUT=a
//SYSLST    DD   SYSOUT=a      AUDIT REPORT
//SYSOUT    DD   SYSOUT=a      INTEGRITY ANALYSIS REPORT
//SNAPIT    DD   SYSOUT=a
//SYSUDUMP  DD   SYSOUT=a
//*
//*FILES NEEDED FOR AUDITING OR FIXING
//*INTEGRATED INDEX SETS.
//*
//DBKYWORK  DD   DSN=dbaudit.dbkywork,
//               DISP=OLD
//INDXEXTR  DD   DSN=dbaudit.indxextr,
//               DISP=OLD
//INDXWORK  DD   DSN=dbaudit.indxwork,
//               DISP=OLD
//*
//*FILES NEEDED ONLY IF DISK OPTION IS SPECIFIED
//*AND VSAM IS CHOSEN
//*
//DBKHASH   DD   DSN=vsam.dbkhash,
//               DISP=SHR
//DBKSAVE   DD   DSN=vsam.dbksave,
//               DISP=SHR
//*
//*THE INPUT STATEMENTS SHOWN HERE ARE EXAMPLES
//*FOR AUDIT AND SIMULATED FIX
//*
//SYSIPT    DD   *
  PROCESS,SUBSCHEMA=subschema-name,
   READYMODE=RETRIEVAL,SORTMSGS=CRITICAL
  AUDIT,PAGES,USERSETS,CALCSETS,INDEX,
   CORE,SETLIMIT=3000,DBKEYTBL
  FIX=SIMULATE,PAGES,USERSETS,CALCSETS,INDEXORPHANS
  REPORTS=YES,RECDATA=HEX,SETMBRS=ALL
/*
//SYSIDMS   DD   *
  SYSIDMS PARAMETERS...
```

Figure 5.9b: OS/390 JCL to Run CA-IDMS/DB Audit

```
/*
//*       FILES NEEDED ONLY IF DISK OPTION IS SPECIFIED
//*       AND BDAM IS CHOSEN
//*
//DBKHASH  DD   DSN=dbaudit.dbkhash,
//              DISP=SHR
//              DCB=DSORG=DA
//DBKHA@@  DD   DSN*.DBKHASH,VOL=REF=*.DBKHASH,
//              DISP=SHR
//              DCB=DSORG=DA
//DBKSAVE  DD   DSN=dbaudit.dbksave,
//              DISP=SHR
//              DCB=DSORG=DA
//DBKSA@@  DD   DSN*.DBKSAVE,VOL=REF=*.DBKSAVE,
//              DISP=SHR
//              DCB=DSORG=DA
```

Figure 5.9c: OS/390 JCL to Run CA-IDMS/DB Audit

5.10.1.5 Key to Sample OS/390 JCL to Run CA-IDMS/DB Audit

#K — The number of kilobytes of storage to be made available to the program execution phase. See 5.8, “Storage Requirements.”

your.loadlib — The dataset name of the MVS/PDS library into which you downloaded CA-IDMS/DB Audit.

idms.loadlib — The dataset name of the MVS/PDS library in which your DMCL and subschema reside.

sort.loadlib — The dataset name of the MVS/PDS library in which your sort modules reside.

disk — The unit address of the disk drive you want to use for the CA-IDMS/DB Audit execution file, or a generic assignment to indicate a disk drive allocation.

ps,ss — Primary/secondary space allocation for the SORT. Space allocation depends on the number of error extract records created in the audit phase. In most cases, the number of extracts created is small. A primary/secondary specification of **5,2** should be adequate. The total sortwork space allocated should be twice the space allocated to the EXTRACTS file.

If your database has numerous errors or the sets containing errors are long, the extract file may be large. In that case, you may need more than four SORTWK datasets.

If you are auditing integrated index sets and you have index orphans, then you need enough extra space to accommodate twice the number of index orphans written to the index extract file.

dbfil01/dbfilnn — The DDNAMEs of the database files to be accessed.

user.dbfil01/user.dbfilnn — The dataset names of the database files to be accessed.

dbaudit.extract — The dataset name assigned to your extract file.

dbaudit.workfile — The dataset name assigned to your work file.

sortmsg — The DDNAME of the message file for your SORT product.

a — The appropriate SYSOUT class for your installation.

dbaudit.dbkywork — The dataset name assigned to your db-key work file. This file is needed only if you are auditing or fixing integrated index sets, and you have mandatory-automatic integrated index sets.

dbaudit.indxextr — The dataset name assigned to your integrated index extract file. This file is needed only if you are auditing or fixing integrated index sets.

dbaudit.indxwork — The dataset name assigned to your integrated index work file. This file is needed only if you are auditing or fixing integrated index sets.

vsam.dbkhash — The dataset name of the VSAM file for the db-key hash table. This file is needed only if you specify DISK in the AUDIT statement. You must predefine this file using the IDCAMS control statements shown in Figure 5.5. Optionally, this file may be defined as BDAM (see Figures 5.7c, 5.8a, and 5.9c).

vsam.dbksave — The dataset name of the VSAM file for the db-key save table. This file is needed only if you specify DISK in the AUDIT statement. You must predefine this file using the IDCAMS control statements shown in Figure 5.5. Optionally, this file may be defined as BDAM (see Figures 5.7c, 5.8a, and 5.9c).

subschema-name — The name of the subschema that describes the database you want audited.

SYSIDMS — This file is always required for the CA-IDMS physical environment parameter to be read. For instance, DMCL=**xxxxxxxx** must be specified, where **xxxxxxxx** is the name of the specific DMCL. See the *CA-IDMS Release 15.0 User Guide* for more information.

dbaudit.dbkhash — The dataset name assigned to your BDAM db-key hash table. This file is only needed if you specify the DISK option of the AUDIT statement.

dbaudit.dbksave — The dataset name assigned to your BDAM db-key save table. This file is only needed if you specify the DISK option of the AUDIT statement.

Figure 5.10: Key to Figure 5.9

5.10.2 Restarting CA-IDMS/DB Audit

When an audit and simulated fix have been successfully completed, you can restart CA-IDMS/DB Audit and actually update the database. The JCL for the update run is the same as the previous JCL, except for the parameter statements. Be sure that no other updates are applied to the database between the audit run and the fix run.

Here are sample parameter statements:

```
PROCESS,SUBSCHEMA=subschema-name,
    READYMODE=UPDATE,SORTMSGS=CRITICAL
FIX=UPDATE,PAGES,USERSETS,CALCSETS,INDEXORPHANS
REPORTS=YES,RECDATA=CHAR,SETMBRS=ERRSONLY
```

5.10.3 Obtaining Reports for a Previous Run

You may want only the reports for a previous run of CA-IDMS/DB Audit. Again, you can use the same JCL and change only the parameter statements:

```
PROCESS,SUBSCHEMA=subschema-name,
    READYMODE=RETRIEVAL,SORTMSGS=CRITICAL
REPORTS=YES,RECDATA=HEX,SETMBRS=ALL
```

5.11 VSE/ESA Operations

Sample JCL for using CA-IDMS/DB Audit in a VSE/ESA environment is presented below in two parts:

1. (Optional) Allocating work files for the DISK option
2. Running CA-IDMS/DB Audit

An explanation of various restart options, showing the use of files created in a previous audit run, follows the JCL.

5.11.1 VSE/ESA File Assignments

Even if you use a storage management tool such as CA-DYNAM, CA-IDMS/DB Audit requires an ASSGN statement for every file except SORTWK*nn*. This ASSGN is required because CA-IDMS/DB Audit has its own device-independent support which dynamically builds a DTF based on the device type indicated by the ASSGN. Unless the ASSIGN specifies VSAM or BDAM, the file may be defined with either DLBL or TLBL.

5.11.2 VSE/ESA JCL

5.11.2.1 1. Allocating Work Files for the DISK Option

(USADEFN) If you specify DISK in the Audit statement, you must allocate two VSAM files to disk: one for the db-key table and one for the db-key save table. See 5.9.1, “Allocating Work Files for the DISK Option” and Figure 5.5 above for sample IDCAMS control statements for file allocation.

5.11.2.2 2. Running CA-IDMS/DB Audit

Sample VSE/ESA JCL to audit and simulate fixing the database described by your subschema is shown in Figures 5.11a and 5.11b. A key to the variables (shown in **bold**) is shown in Figures 5.12a and 5.12b. Similar JCL is contained in TOOLJCL library member USAEXEC.

```
// JOB DBAUDIT
// OPTION LOG,PARTDUMP
*
*  ────────── VSAM WORK FILE'S FOR AUDIT DISK OPTION ───────
*
// DLBL   DBKHASH,'vsam.dbkhash',0,VSAM
// EXTENT SYS014,volser,,,rel-trk-blk,amount
// ASSGN  SYS014,DISK,VOL=volser,SHR
// DLBL   DBKSAVE,'vsam.dbksave',0,VSAM
// EXTENT SYS015,volser,,,rel-trk-blk,amount
// ASSGN  SYS015,DISK,VOL=volser,SHR
// EXEC  IDCAMS,SIZE=AUTO
*
*   See source library member USADEFN for IDCAMS control statements
*
/*
*       ───────────── SORT FILES ─────────────────────
*
// DLBL SORTWK1,'SORT.WORK1',0,SD       SORT WORK #1
// EXTENT SYS001,volser,,,rel-trk-blk,amount
// ASSGN  SYS001,DISK,VOL=volser,SHR
// DLBL SORTWK2,'SORT.WORK2',0,SD       SORT WORK #2
// EXTENT SYS002,volser,,,rel-trk-blk,amount
// ASSGN  SYS002,DISK,VOL=volser,SHR
// DLBL SORTWK3,'SORT.WORK3',0,SD       SORT WORK #3
// EXTENT SYS003,volser,,,rel-trk-blk,amount
// ASSGN  SYS003,DISK,VOL=volser,SHR
// DLBL SORTWK4,'SORT.WORK4',0,SD       SORT WORK #4
// EXTENT SYS004,volser,,,rel-trk-blk,amount
// ASSGN  SYS004,DISK,VOL=volser,SHR
*
*       ───────────── USER DATABASE FILES ──────────
*
// DLBL   dbfil01,'user.dbfil01',,DA    USER DATADASE FILE #01
// EXTENT SYSnnn,volser
// ASSGN  SYSnnn,DISK,VOL=volser,SHR
   .
   .
   .
// DLBL   dbfilnn,'user.dbfilnn',,DA    USER DATADASE FILE #nn
// EXTENT SYSnnn,volser
// ASSGN  SYSnnn,DISK,VOL=volser,SHR
*
*  ────────── VSAM WORK FILE'S FOR AUDIT DISK OPTION ───────
*
// DLBL   DBKHASH,'vsam.dbkhash',0,VSAM
// EXTENT SYS010,volser
// ASSGN  SYS010,DISK,VOL=volser,SHR
// DLBL   DBKSAVE,'vsam.dbksave',0,VSAM
// EXTENT SYS011,volser
// ASSGN  SYS011,DISK,VOL=volser,SHR
*
```

Figure 5.11a: VSE/ESA JCL to Run CA-IDMS/DB Audit

```
*   ———————————— DBAUDIT WORK FILES ————————————————————
*
// ASSGN  SYS020,DISK,VOL=volser,SHR
// DLBL   AFILE04,'AUDIT.extract',0,SD       EXTRACT OUTPUT FILE
// EXTENT SYS020,volser,1,0,rel-trk-blk,amount
// DLBL   AFILE05,'AUDIT.extract',0,SD       EXTRACT INPUT FILE
// EXTENT SYS020,volser,1,0,rel-trk-blk,amount
// ASSGN  SYS021,DISK,VOL=volser,SHR
// DLBL   AFILE06,'AUDIT.workfile',0,SD      WORK OUTPUT FILE
// EXTENT SYS021,volser,1,0,rel-trk-blk,amount
// DLBL   AFILE07,'AUDIT.workfile',0,SD      WORK INPUT FILE
// EXTENT SYS021,volser,1,0,rel-trk-blk,amount
// ASSGN  SYS022,DISK,VOL=volser,SHR
// DLBL   AFILE08,'AUDIT.indxwork',0,SD      INDEX WORK INPUT FILE
// EXTENT SYS022,volser,1,0,rel-trk-blk,amount
// DLBL   AFILE09,'AUDIT.indxwork',0,SD      INDEX WORK OUTPUT  FILE
// EXTENT SYS022,volser,1,0,rel-trk-blk,amount
// ASSGN  SYS023,DISK,VOL=volser,SHR
// DLBL   AFILE10,'AUDIT.indxextr',0,SD      INDEX EXTR INPUT FILE
// EXTENT SYS023,volser,1,0,rel-trk-blk,amount
// DLBL   AFILE11,'AUDIT.indxextr',0,SD      INDEX EXTR OUTPUT  FILE
// EXTENT SYS023,volser,1,0,rel-trk-blk,amount
// ASSGN  SYS024,DISK,VOL=volser,SHR
// DLBL   AFILE12,'AUDIT.dbkywork',0,SD      DBKEY INPUT FILE
// EXTENT SYS024,volser,1,0,rel-trk-blk,amount
// DLBL   AFILE13,'AUDIT.dbkywork',0,SD      DBKEY OUTPUT FILE
// EXTENT SYS024,volser,1,0,rel-trk-blk,amount
// ASSGN  SYSIPT,SYSRDR                       INPUT PARAMETER FILE
// ASSGN  SYS006,SYSLST                       AUDIT REPORT
// ASSGN  SYS007,uuu                          INTEGRITY ANALYSIS REPORT
*
*   ———————————————   LIBRARY DEF'S ————————————————
*
// DLBL TOOL,'tool.loadlib'
// EXTENT ,volser
// DLBL IDMS,'idms.loadlib'
// EXTENT ,volser
// LIBDEF PHASE,SEARCH=(TOOL.sublib,IDMS.sublib)
*
// DLBL SYSIDMS,'#SYSIPT',0,SD
// EXEC   USADRVR,SIZE=(AUTO,#K),PARM='NOSPIE/'
*
*  R14.1 OPTIONAL SYSIDMS PARAMETERS
*
/*
  PROCESS,SUBSCHEMA=subschema-name,....
  AUDIT,.......
  FIX=SIMULATE,.........
/&
* $$ EOJ
```

Figure 5.11b: VSE/ESA JCL to Run CA-IDMS/DB Audit

5.11.2.3 Key to Sample VSE/ESA JCL to Run CA-IDMS/DB Audit

vsam.dbkhash — The data set name of the VSAM file for the db-key table. This file is needed only if you specify DISK in the AUDIT statement. You must pre-define this file by using the IDCAMS control statements.

volser — The volume serial number or generic assignment of the disk volume on which the file, as specified in the preceding DLBL statement, resides.

rel-trk-blk — Relative track or relative block number: the starting position on the DASD for storage of the work file specified in the previous statement.

amount — The space allocation you need to store the file specified in the previous statement. Refer to the specific information on estimating space earlier in this chapter.

vsam.dbksave — The data set name of the VSAM file for the db-key save table. This file is needed only if you specify DISK in the AUDIT statement. You must pre-define this file by using the IDCAMS control statements.

user.dbfil01/user.dbfilnn — The file ID of each of the database files to be accessed.

SYSnnn — The programmer logical units of the database files to be accessed.

extract — The data set name assigned to your extract file.

workfile — The data set name assigned to your work file.

indxwork — The data set name assigned to your integrated index work file. This file is needed only if you are auditing or fixing integrated index sets.

indxextr — The data set name assigned to your integrated index extract file. This file is needed only if you are auditing or fixing integrated index sets.

dbkywork — The data set name assigned to your db-key work file. This file is needed only if you are auditing or fixing integrated index sets, and you have mandatory-automatic integrated index sets.

uuu — SYS007 must be assigned to a second SYSLST device (**uuu**).

tool.loadlib — The data set name of the core image library in which your DMCL and subschema reside.

tool.sublib/idms.sublib — The sublibrary name of the VSE/ESA library specified in the previous file name.

subschema-name... — The parameters you specify to execute CA-IDMS/DB Audit.

Figure 5.12: Key to Figure 5.11

5.11.3 Restarting CA-IDMS/DB Audit

When an audit and simulated fix have been successfully completed, you can restart CA-IDMS/DB Audit and actually update the database. The JCL for the update run is the same as the previous JCL, except for the parameter statements. Be sure that no other updates are applied to the database between the audit run and the fix run.

Here are sample parameter statements:

```
PROCESS,SUBSCHEMA=subschema-name,
    READYMODE=UPDATE,SORTMSGS=CRITICAL
FIX=UPDATE,PAGES,USERSETS,CALCSETS,INDEXORPHANS
REPORTS=YES,RECDATA=CHAR,SETMBRS=ERRSONLY
```

5.11.4 Obtaining Reports for a Previous Run

You may want only the reports for a previous run of CA-IDMS/DB Audit. Again, you can use the same JCL and change only the parameter statements:

```
PROCESS,SUBSCHEMA=subschema-name,
    READYMODE=RETRIEVAL,SORTMSGS=CRITICAL
REPORTS=YES,RECDATA=HEX,SETMBRS=ALL
```

5.12 VM/ESA Operations

Sample EXEC for using CA-IDMS/DB Audit in a VM/ESA environment is shown below in two parts:

1. (Optional) Allocating work files for the DISK option
2. Running CA-IDMS/DB Audit

An explanation of various restart options, showing the use of files created in a previous audit run, follows the EXEC.

5.12.1 VM/ESA EXEC

5.12.1.1 1. Allocating Work Files for the DISK Option

(USADEFN) If you specify DISK in the Audit statement, you must allocate two direct access disk files: one for the db-key hash table and one for the db-key save table. These direct access files may be VSAM or BDAM. If you choose to allocate these files as VSAM, they must be defined through AMSERV. See 5.9.1, “Allocating Work Files for the DISK Option” and Figure 5.5 above for sample AMSERV control statements for file allocation.

If you choose to allocate these files as BDAM, you do not need to use AMSERV. Instead, revise your EXEC (as shown in Figure 5.13b) to contain the statements shown in Figure 5.13a.

```
* FILES NEEDED ONLY IF DISK OPTION IS SPECIFIED
* AND BDAM IS CHOSEN
*
FILEDEF   DBKHASH       fn ft fm6 (DSORG DA
FILEDEF   DBKHA@@       fn ft fm6 (DSORG DA
FILEDEF   DBKSAVE       fn ft fm6 (DSORG DA
FILEDEF   DBKSA@@       fn ft fm6 (DSORG DA
*
* DBKHA@@ MUST HAVE THE SAME FILEID AS DBKHASH
* AND DBKSA@@ MUST HAVE THE SAME FILEID AS DBKSAVE
```

Figure 5.13a: VM/ESA EXEC to Run CA-IDMS/DB Audit

5.12.1.2 2. Running CA-IDMS/DB Audit

Sample VM/ESA EXEC to audit and simulate fixing the database described by your subschema is shown in Figure 5.13a and 5.13b. A key to the variables (shown in **bold**) is shown in Figures 5.14. A similar EXEC is contained in the source library member USAEXEC.

```
TOOL_LOADLIB_FN     = 'toollib'
IDMS_LOADLIB_FN     = 'idmslib'
SORTLIB_FN          = 'sortlib'
/*
/* Link and access the Minidisks containing the required librarie(s)
/* and database file(s).
/*
 'CP SPOOL PRINTER NOCONT CLOSE'
 'CP SPOOL PRINTER TO * NOHOLD CONT FORM OFF DIST OFF'
 'GLOBAL  LOADLIB ' TOOL_LOADLIB_FN IDMS_LOADLIB_FN
 'GLOBAL  TXTLIB ' SORTLIB_FN
/*
/* Insert FILEDEFS as needed to define all required database files.
/*
 'FILEDEF dbfil01 DISK fn ft fm6 ( dcb'
 'FILEDEF dbfilnn DISK fn ft fm6 ( dcb'
/*
/* Files needed for all runs.
/*
 'FILEDEF SYSLST   PRINTER'
 'FILEDEF SYSUDUMP PRINTER'
 'FILEDEF SYSOUT   PRINTER'
 'FILEDEF SNAPIT   PRINTER'
 'FILEDEF EXTRACTS DISK USAEXEC EXTRACTS fm'
 'FILEDEF WORKFILE DISK USAEXEC WORKFILE fm'
/*
/* You must create a file 'SYSIDMS PARMS A' containing the SYSIDMS
/* parameters you use to specify your runtime environment.
/*
 'FILEDEF SYSIDMS  DISK SYSIDMS PARMS A'
/*
/* You must create a file 'USAEXEC SYSIPT A' containing the input
/* parameter statements prior to executing this EXEC.
/*
 'FILEDEF SYSIPT   DISK USAEXEC SYSIPT   A'
/*
/* Insert FILEDEF statements for SORT work space as required by
/* your SORT product.
/*
/* Files needed for auditing or fixing integrated index sets.
/*
 'FILEDEF DBKYWORK DISK USAEXEC DBKYWORK fm'
 'FILEDEF INDXEXTR DISK USAEXEC INDXEXTR fm'
 'FILEDEF INDXWORK DISK USAEXEC INDXWORK fm'
/*
/* The remaining VSAM files are needed only if you specify DISK
/* option on the "AUDIT" parameter statement and VSAM is chosen.
/*
/* Remove the comments from around the next 4 statements.
/*      'DLBL IJSYSCT vsam_fm DSN vsamcat'
        'DLBL IJSYSUC vsam_fm DSN vsamcat'
        'DLBL DBKSAVE vsam_fm DSN dbksave (VSAM'
        'DLBL DBKHASH vsam_fm DSN dbkhash (VSAM'
 */
```

Figure 5.13b: VM/ESA EXEC to Run CA-IDMS/DB Audit

```
SAY 'START DATABASE AUDIT"
SIGNAL OFF ERROR
 'EXECOS OSRUN USADRVR PARM=''NOSPIE/'''
 USADRVR_RC = RC
IF USADRVR_RC > 4
 THEN DO
  CALL ERROR
 END
 'CP SPOOL PRINTER NOCONT'
 'CP CLOSE PRINTER NAME DBAUDIT LISTING'
 'CP SPOOL PRINTER OFF'
 SAY 'USAEXEC FINISHED WITH A RETURN CODE OF' USADRVR_RC
 'GLOBAL LOADLIB'
 'GLOBAL TXTLIB'
 'FILEDEF * CLEAR'
 EXIT USADRVR_RC
/*
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
ERROR:
/*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 ERROR_RC = RC
 TRACE OFF; SIGNAL OFF ERROR
 SAY 'NON-ZERO RETURN CODE ENCOUNTERED IN EXEC AT LINE' SIGL
'CP SPOOL PRINTER NOCONT'
'CP CLOSE PRINTER NAME DBAUDIT LISTING'
'CP SPOOL PRINTER OFF'
 'GLOBAL LOADLIB'
 'GLOBAL TXTLIB'
 'FILEDEF * CLEAR'
 EXIT ERROR_RC
/*
```

Figure 5.13b (cont.): VM/ESA EXEC to Run CA-IDMS/DB Audit

5.12.1.3 Key to Sample VM/ESA EXEC to Run CA-IDMS/DB Audit

toollib — The file name of the load library into which you downloaded CA-IDMS/DB Audit.

idmslib — The file name of the load library containing your CA-IDMS subschema and DMCL modules.

sortlib — The file name of the text library containing your sort modules.

dbfil01/dbfilnn — The name of each of the database files to be accessed.

fn ft fm — The file name, file type, and file mode of the relevant file. The file mode number of your database file(s) must be 6.

dcb — The data control block (DCB) information for the files.

Note: The following definitions are required if you specify the DISK option in the AUDIT statement. You must predefine these files using IDCAMS control statements. See Chapter 5, “Operations” of the CA-IDMS/DB Audit User Guide.

dbksave — The data set name of the VSAM file for the db-key save table.

dbkhash — The data set name of the VSAM file for the db-key hash table.

vsamcat — The data set name of the VSAM catalog.

vsam_fm — The file mode of the accessed minidisk containing your VSAM catalog(s) and file(s).

Figure 5.14: Key to Figure 5.13a

5.12.2 Restarting CA-IDMS/DB Audit

When an audit and simulated fix have been successfully completed, you can restart CA-IDMS/DB Audit and actually update the database. The EXEC for the update run is the same as the previous EXEC, except for the parameter statements. Be sure that no other updates are applied to the database between the audit run and the fix run.

Here are sample parameter statements:

```
PROCESS,SUBSCHEMA=subschema-name,
    READYMODE=UPDATE,SORTMSGS=CRITICAL
FIX=UPDATE,PAGES,USERSETS,CALCSETS,INDEXORPHANS
REPORTS=YES,RECDATA=CHAR,SETMBRS=ERRSONLY
```

5.12.3 Obtaining Reports for a Previous Run

You may want only the reports for a previous run of CA-IDMS/DB Audit. Again, you can use the EXEC and change only the parameter statements:

```
PROCESS,SUBSCHEMA=subschema-name,
    READYMODE=RETRIEVAL,SORTMSGS=CRITICAL
REPORTS=YES,RECDATA=HEX,SETMBRS=ALL
```

Chapter 6. Messages

This chapter lists all messages that appear on the CA-IDMS/DB Audit reports. The messages are listed in order, by code, along with reasons for their occurrences and suggested actions.

CA-IDMS/DB Audit messages are displayed as **xxxxnnns**. The **xxxx** is a four character abbreviation of the module that generated the message, **nnn** is a three digit code, and **s** is the severity class. In this chapter, messages are listed without the module abbreviation.

As CA-IDMS/DB Audit performs its various processes, it records three classes of messages on the Audit Report.

- Error
- Informative
- Warning

6.1.1 Error Messages

Error messages are suffixed by an nnnE message number and report on the following:

- Erroneous parameter data
- Conflicting parameter data
- Irrecoverable processing conditions
- Physical integrity errors

6.1.2 Informative Messages

Informative messages are suffixed by an nnnI message number and report on the following:

- Parameter information
- Start and stop of each processing phase
- Actions taken
- Statistical information
- Physical integrity errors

6.1.3 Warning Messages

Warning messages are suffixed by a nnnW message number and report on the following:

- Parameter information
- Actions taken

6.2 Error Messages

CA-IDMS/DB Audit error messages are suffixed by an *nnn*E number and include possible causes for the message and action to be taken to correct the problem.

001E NOT ENOUGH STORAGE FOR PROCESSING. AMOUNT OF CONTIGUOUS STORAGE REQUIRED = number-of-bytes BYTES

Reason: The amount of available storage in the region or partition was not large enough for CA-IDMS/DB Audit's pseudo subschema table or other work areas. The required number of contiguous bytes of storage is displayed.

Action: See the “Operations” chapter for more details about CA-IDMS/DB Audit storage requirements. Increase the GETMAIN/GETVIS area by changing the REGION/SIZE parameter on the EXEC statement or lower the SETLIMIT value so that less storage is required.

002E KEYWORD PROCESSOR TABLE (USAT0010) MISSING

Reason: Module USAT0010 was not link edited with CA-IDMS/DB Audit. CA-IDMS/DB Audit terminates processing.

Action: Contact Computer Associates Product Support for assistance.

003E UNRECOGNIZABLE PARAMETER VALUE value

Reason: An unrecognizable parameter value was returned by the keyword processor. CA-IDMS/DB Audit terminates abnormally.

Action: Contact Computer Associates Product Support for assistance.

005E EXPECTED CONTINUATION STATEMENT NOT RECEIVED

Reason: An input parameter statement containing a comma followed the last keyword phrase when end-of-file was reached on SYSIPT.

Action: If continuation is desired, add an additional parameter statement to the parameter file.

If continuation is <u>not</u> desired, delete the comma after last keyword phrase. Resubmit the job.

006E MAJOR KEYWORD NOT FOUND

Reason: The first word in the parameter statement was not a major keyword.

Action: Check parameter statement for misspelled keywords. Correct the keywords and resubmit the job.

007E UNRECOGNIZABLE KEYWORD keyword

Reason: An unrecognizable major or minor keyword was detected.

Action: Check parameter statement for misspelled or invalid keywords. Correct the keywords and resubmit the job.

008E INVALID FORMAT FOR KEYWORD keyword

Reason: The designated keyword was incorrectly specified. Absence of a keyword will not invoke this message.

Action: Check the parameter statement for correct parameter syntax. Correct the keyword and resubmit the job.

009E KEYWORD VALUE TOO SMALL/LARGE keyword

Reason: Value for the designated keyword was smaller or larger than the value allowed.

Action: Correct the keyword value and resubmit the job.

010E PROCESS PARAMETER MISSING OR INVALID

Reason: PROCESS parameter not detected as first non-comment parameter statement. PROCESS keyword misspelled.

Action: Check the parameter file and make sure the PROCESS statement is the first non-comment parameter statement. Correct spelling of the keyword and resubmit the job.

011E MULTIPLE OCCURRENCES OF SAME PARAMETER STATEMENT

Reason: More than one PROCESS, AUDIT, FIX or REPORTS parameter statement encountered. Multiple AREA parameters specifying identical area name encountered. Multiple RECORD parameters specifying identical record name encountered. Multiple SET parameters specifying identical set name encountered.

Action: Delete all but one occurrence of multiple parameters and resubmit job. Delete extra parameter statements or change area, set, or record name on duplicate parameter statements. Resubmit the job.

012E NULL VALUE NOT VALID FOR KEYWORD keyword

Reason: A required value was not entered for the designated keyword.

Action: Specify a value for the keyword and resubmit the job.

013E INVALID VALUE SPECIFIED FOR KEYWORD keyword

Reason: An invalid value was specified for the designated keyword.

Action: Correct the keyword value and resubmit the job.

014E NUMERIC VALUE NOT SPECIFIED FOR KEYWORD keyword

Reason: A non-numeric value was specified for the designated keyword.

Action: Correct the keyword value and resubmit the job.

015E AUDIT,CALCSETS SPECIFIED BUT SR1 RECORD/CALC SET IS NOACCESS/NOAUDIT

Reason: Auditing of the CALC set cannot be performed because a RECORD or SET statement, or both, contained NOACCESS.

Action: Delete NOACCESS from RECORD and SET statements and resubmit the job.

016E AUDIT,CALCSETS SPECIFIED BUT SR1 RECORD IS NOSWEEP

Reason: Auditing of CALC set cannot be performed because a RECORD statement contained NOSWEEP.

Action: Delete NOSWEEP from the RECORD statement and resubmit the job.

017E AREA/RECORD/SET PARAMETER ENTERED WITHOUT AUDIT PARAMETER

Reason: An AREA, RECORD, or SET statement was entered without also entering an AUDIT statement.

Action: Enter an AUDIT statement and resubmit the job. If only fixing or reporting is to be performed, delete the AREA, RECORD, or SET statement.

018E AREA/RECORD/SET NOT IN SUBSCHEMA

Reason: Incorrect subschema specified. Area name, record name, or set name was misspelled.

Action: Check that the correct subschema was specified and that subschema name is spelled correctly. Resubmit job. Correct area name, record name, or set name and resubmit the job.

019E BEGIN/END PAGE NUMBER OUTSIDE AREA PAGE RANGE

Reason: The specified BEGIN or END page number in an AREA statement is outside the area page range or a null value was entered. An incorrect area name was entered in an AREA statement.

Action: Correct the page number value and resubmit the job. Correct the area name and resubmit the job.

020E BEGIN PAGE NUMBER GREATER THAN END PAGE NUMBER

Reason: The specified BEGIN page number is greater than the specified END page number in an AREA statement.

Action: Correct the BEGIN and END page numbers and resubmit the job.

022E FIX OPERAND SPECIFIED WITH NO CORRESPONDING AUDIT OPERAND

Reason: A fix option was specified for a portion of the database for which there was no corresponding audit specified.

Action: Correct the fix request and resubmit the job.

023E ALL AREAS IN SUBSCHEMA ARE NOACCESS

Reason: Auditing cannot be performed because all areas in the subschema were designated NOACCESS in AREA parameter statements.

Action: Delete NOACCESS from at least one AREA statement and resubmit the job.

024E ALL SETS IN SUBSCHEMA ARE NOAUDIT

Reason: Auditing cannot be performed because all sets in the subschema were designated NOACCESS in either AREA, RECORD, or SET parameter statements.

Action: Delete NOACCESS from the appropriate AREA, RECORD, or SET parameter statements and resubmit the job.

025E ready-mode NOT ALLOWED FOR AREA area-name

Reason: The designated area cannot be audited because the area has a privacy lock of NO for Ready Exclusive Update or Ready Shared Retrieval.

Action: Specify another subschema that allows the area to be readied in the appropriate mode and resubmit the job. Designate the area as NOACCESS using an AREA parameter statement and resubmit the job.

026E FIND/GET NOT ALLOWED FOR record-name

Reason: The designated record cannot be audited because it has a privacy lock of NO for the DML command FIND or GET in the subschema being used.

Action: Specify another subschema that allows FIND and GET for all records to be audited. Designate the record as NOACCESS, by using a RECORD parameter statement. Resubmit the job.

027E MODIFY NOT ALLOWED FOR RECORD record-name

Reason: The designated record cannot be fixed because it has a privacy lock of NO for the DML command MODIFY in the subschema being used.

Action: Specify another subschema that allows MODIFY for all records to be audited. Designate the record as NOACCESS, by using a RECORD parameter statement during the audit phase. Rerun the audit and fix phases.

028E ERASE NOT ALLOWED FOR RECORD record-name

Reason: When LDEL record fixing was requested, the designated record could not be fixed because it had a privacy lock of NO for the DML command ERASE in the subschema being used.

Action: Specify another subschema that allows ERASE for all records to be audited. Designate the record as NOACCESS, BY using a RECORD parameter statement during the audit phase. Rerun the audit and fix phases.

029E RECORD ID record-id NOT IN SUBSCHEMA FOR MULTI-MEMBER SET set-name

Reason: The indicated member record was not copied into subschema being used for the indicated set.

Action: Specify another subschema that has copied the member record and resubmit the job. Designate the set as NOACCESS by using a SET statement and resubmit the job.

030E DISCONNECT SPECIFIED FOR NOACCESS RECORD record-name

Reason: The DISCONNECT option was specified for a record that was designated as NOACCESS in either an AREA or RECORD parameter statement.

Action: Delete the DISCONNECT option and resubmit the job. Delete the NOACCESS option and resubmit the job.

031E DISCONNECT SPECIFIED FOR RECORD record-name IN NOAUDIT SET set-name

Reason: The DISCONNECT option was specified for a member record of a set designated as NOACCESS in either an AREA, RECORD, or SET statement.

Action: Delete the DISCONNECT option and resubmit the job. Delete the NOACCESS option and resubmit the job.

032E DISCONNECT SPECIFIED FOR NOAUDIT SET set-name

Reason: The DISCONNECT option was specified for a set designated as NOACCESS in either an AREA, RECORD, or SET statement.

Action: Delete DISCONNECT option and resubmit the job. Delete the NOACCESS option and resubmit the job.

033E DISCONNECT SPECIFIED FOR RECORD NOT A MEMBER OF ANY SETS

Reason: The DISCONNECT option was specified in a RECORD statement for a record that is not a member of a set.

Action: Delete the DISCONNECT option from the RECORD statement and resubmit the job.

034E NO AUDIT TYPE SPECIFIED ON AUDIT PARAMETER

Reason: No options were specified in the AUDIT statement to indicate the type of auditing to be performed.

Action: Add the desired options to the AUDIT statement and resubmit the job.

035E DBKEYTBL/WALKNEXT SPECIFIED WITHOUT ALCSETS/USERSETS

Reason: DBKEYTBL and/or WALKNEXT was requested in an AUDIT statement without CALCSETS or USERSETS.

Action: Add CALCSETS and/or USERSETS to the AUDIT statement and resubmit the job. Delete DBKEYTBL and WALKNEXT from the AUDIT statement and resubmit the job.

036E NO FIX TYPE SPECIFIED ON FIX PARAMETER

Reason: No type of fixing was specified in the FIX statement.

Action: Add the desired fix options to the FIX statement and resubmit the job.

037E FIX SUBSCHEMA DOES NOT MATCH AUDIT SUBSCHEMA

Reason: Fixing was requested using an extract file created by a previous audit job. The subschema name used by the audit phase, however, does not match the subschema name specified in the PROCESS statement of the fix phase.

Action: Correct the subschema name in the PROCESS statement to match the subschema name in the extract file of the audit phase. Resubmit the job.

038E FIX NOT ALLOWED   AUDIT PHASE ABORTED

Reason: Fix processing cannot be performed because of an irrecoverable error during the audit phase.

Action: Review messages displayed during the audit phase and correct any errors. Rerun both the audit and fix phases.

039E FIX NOT ALLOWED STOPAFTER LIMIT EXCEEDED IN AUDIT PHASE

Reason: Fix processing cannot be performed because the STOPAFTER limit was exceeded during the audit phase.

Action: Increase the value of STOPAFTER. Rerun both the audit and fix phases.

040E FIX NOT ALLOWED SETLIMIT VALUE EXCEEDED IN AUDIT PHASE

Reason: Fix processing cannot be performed because the SETLIMIT value was exceeded during the audit phase.

Action: If the DBKEYTBL option was not requested, a set loop condition may exist. Resubmit both the audit and fix phases requesting the DBKEYTBL option. If DBKEYTBL was requested, increase the value of SETLIMIT. Rerun both the audit and fix phases.

041E UNSUCCESSFUL OPEN OF file-name FILE

Reason: CA-IDMS/DB Audit could not successfully open the designated file. Processing is terminated.

Action: Check that all file names are correctly spelled. Add the appropriate file name to the JCL. Resubmit the job.

042E EXTRACT RECORD MISSING OR INVALID

Reason: The audit phase that created the extract file was canceled before the header or trailer record was written. CA-IDMS/DB Audit terminates processing. The file being read by the report or fix phase is not a CA-IDMS/DB Audit extract file.

Action: Rerun the audit and fix phases. Resubmit the job using the correct extract file.

043E MULTIPLE EXTRACT HEADER/TRAILER RECORDS DETECTED

Reason: Extract files from multiple CA-IDMS/DB Audit jobs were concatenated. CA-IDMS/DB Audit terminates processing.

Action: Resubmit the job using only one extract file.

044E UNSUCCESSFUL BIND RUN-UNIT

Reason: A non-zero error status was returned when a BIND RUN-UNIT command was issued. CA-IDMS/DB Audit terminates processing.

Action: See message 050I.

045E UNSUCCESSFUL BIND record-name/page-number

Reason: A non-zero error status was returned when a BIND command for the designated record or page was issued. CA-IDMS/DB Audit terminates processing.

Action: See message 050I.

046E UNSUCCESSFUL READY ready-mode area-name

Reason: A non-zero error status was returned when the indicated ready mode was invoked for the designated area. CA-IDMS/DB Audit terminates processing.

Action: See message 050I.

047E UNSUCCESSFUL FINISH

Reason: A non-zero error status was returned after a FINISH command. CA-IDMS/DB Audit terminates processing.

Action: See message 050I.

049E UNSUCCESSFUL OBTAIN FIRST/NEXT RECORD/PAGE WITHIN AREA LAST GOOD RECORD=record-name/PAGE=page-number

Reason: A non-zero error status was returned after an OBTAIN command for a first record, a next record, or a page within the area displayed with the preceding message I018. The db-key is displayed for the last record retrieved in the area, or the page number is displayed for the last page retrieved in the area sweep. If the record db-key or the page number equals HIGH-VALUES, an OBTAIN FIRST command was issued; otherwise, an OBTAIN NEXT command was issued. Message 050I follows this message.

Action: See message 050I.

050E UNSUCCESSFUL OBTAIN OF RECORD/PAGE

Reason: A non-zero error status was returned after an OBTAIN command for a record or page. In the audit phase, CA-IDMS/DB Audit terminates processing of the area or set occurrence. In the fix phase, CA-IDMS/DB Audit terminates all processing.

Action: Audit phase--see message 050I. Fix phase--contact Computer Associates Product Support.

051E UNSUCCESSFUL MODIFY OF record-name/page-number

Reason: A non-zero error status was returned when issuing a MODIFY command was issued during the fix phase. CA-IDMS/DB Audit terminates.

Action: Audit phase--see message 050I for the error status. Contact Computer Associates Product Support.

052E UNSUCCESSFUL ERASE OF RECORD

Reason: A non-zero error status was returned when an ERASE command was issued during the fix phase. CA-IDMS/DB Audit terminates.

Action: Contact Computer Associates Product Support.

Note: If errors 050E, 051E, or 052E occur during fix phase, the database must be restored to its state prior to running CA-IDMS/DB Audit. A SNAP of appropriate CA-IDMS/DB Audit work areas will automatically be produced. Contact Computer Associates Product Support for assistance.

053E STOPAFTER LIMIT EXCEEDED

Reason: The STOPAFTER value was exceeded by one physical integrity error. CA-IDMS/DB Audit terminates the audit phase.

Action: Increase the STOPAFTER value and resubmit the job.

054E SETLIMIT EXCEEDED FOR SET set-name

Reason: The SETLIMIT value was exceeded for the designated set. CA-IDMS/DB Audit terminates processing of the set occurrence.

Action: If DBKEYTBL was not requested, a set loop may exist in the set occurrence. Resubmit the job requesting DBKEYTBL. If DBKEYTBL was requested, increase the SETLIMIT value. Resubmit the job.

056E REPORT/FIX PHASE TERMINATED

Reason: The SORT program returned a non-zero return code to the designated CA-IDMS/DB Audit phase. An error was detected that makes execution of the phase impossible.

Action: Review the SORT messages and correct the problem. Resubmit the job. Review the associated error message and correct the problem. Resubmit the job.

057E ORPHANS OR MEMBERS OF ANOTHER SET POINT INTO A SET WHICH WAS NOT AUDITED set-name

Reason: CA-IDMS/DB Audit has performed a partial area audit and one of the following conditions exists:

- A set containing an integrity error points into a second set containing an integrity error and the owner of the second set is outside the BEGIN/END page range for the area where the owner resides.
- Records are detected as orphans of a set containing integrity errors. The owner of the set is outside the BEGIN/END page range for the area where the owner resides. In either case, the indicated set was not audited. Information about the owner of the indicated set occurrence follows this message. If this error is displayed during the fix phase, processing is terminated.

Action: If Error 057E occurred during the fix phase, specify a BEGIN/END page range for the area that includes the owner of the indicated set, and rerun both the audit and fix phases. If the error occurred during the report phase, processing continues. The Before Images Report does not, however, accurately display the sets involved.

058E header/footer PAGE NUMBER DOES NOT MATCH REQUESTED PAGE NUMBER

Reason: One of the following conditions exists.

- The page number within the page header or footer does not match the page number that was read during area sweep processing.
- The free space on the page, as indicated by the line space size in the footer and the available space size in the header, is not all low-values.

Action:

- Review the associated message and ensure that the correct SUBSCHEMA/DMCL and load/core image library are specified before attempting any fixing. Otherwise, a large number of pages can be incorrectly flagged as being in error.
- Run FIX=UPDATE,PAGES to set the free space to low-values.

059E RECORD DBKEY OUTSIDE SPECIFIED FROM/THRU PAGE RANGE

Reason: The db-key of displayed record occurrence does not fall within the from/thru page range specified for record type.

Action: None. CA-IDMS/DB Audit cannot fix this type of error.

Note: This type of error does not affect the walking of a set occurrence. It can, however, cause CALC records not to be found.

060E NEXT/PRIOR/OWNER/INDEX DBKEY IS NULL

Reason: During USERSETS auditing, the displayed record occurrence contained a next, prior, or owner db-key that is null.

Action: Review associated messages and determine if fixing should be performed. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the record resides and patch the bad page.

061E NEXT/PRIOR/OWNER/INDEX DBKEY OUTSIDE AREA PAGE RANGES FOR SUBSCHEMA

Reason: During USERSETS/CALCSETS auditing, the displayed record occurrence contained either a next, prior, or owner db-key that did not point into any of the areas copied into the subschema.

Action: Review associated messages to determine if fixing should be performed. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the record resides and patch the bad page.

062E NEXT/PRIOR/OWNER/INDEX DBKEY POINTS INTO NON-READIED AREA OF SUBSCHEMA

Reason: During USERSETS/CALCSETS auditing, the displayed record occurrence contained either a next, prior, or owner db-key that pointed into an non-readied area of the subschema.

Action: Review associated messages to determine if fixing should be performed. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the record resides and patch the bad page.

063E NEXT/PRIOR/OWNER/INDEX DBKEY POINTS TO NON-EXISTENT RECORD

Reason: During CALCSETS auditing, the displayed record occurrence contained a next, prior, or owner db-key that is null. This error can occur if all CALC records have not been copied into the subschema being used. During USERSETS/CALCSETS auditing, the displayed record occurrence contained either a next, prior, or owner db-key that pointed to a non-existent record.

Action: Be certain that all CALC records have been copied into the subschema being used. Review associated messages to determine if fixing should be performed. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the record resides and patch the bad page.

064E NEXT/PRIOR/OWNER/INDEX DBKEY POINTS TO INVALID RECORD TYPE FOR SET

Reason: During USERSETS/CALCSETS auditing, the displayed record occurrence contained either a next, prior, or owner db-key that pointed to an invalid record type for the set.

Action: Review associated messages to determine if fixing should be performed. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the record resides and patch the bad page.

065E SORT KEY OUT OF SEQUENCE

Reason: When sorted in ascending order, the displayed record's sort key was less than the sort key of the prior record (of the same record type) in the set. When sorted in descending order, the displayed record's sort key was greater than the sort key of the prior record in the set.

Action: Review the associated messages and determine if fixing should be performed.

066E DUPLICATE SORT KEY DETECTED

Reason: The displayed record's sort key was equal to the sort key of the prior record in a set where duplicates were not allowed.

Action: None. CA-IDMS/DB Audit cannot fix this type of error.

067E OWNER DBKEY DOES NOT POINT TO OWNER RECORD OF SET

Reason: Owner db-key of displayed record is invalid. Displayed record is member of another set occurrence pointed into by the current set occurrence.

Action: Review associated messages and determine if fixing should be performed.

068E PRIOR DBKEY DOES NOT POINT BACK TO PRIOR RECORD IN SET

Reason: When walking next in the set occurrence, the prior db-key of the displayed record did not match the db-key of the prior record in the set occurrence.

Action: Review associated messages and determine if fixing should be performed.

069E NEXT DBKEY DOES NOT POINT BACK TO NEXT RECORD IN SET

Reason: When walking prior in the set occurrence, the next db-key of the displayed record did not match the db-key of the next record in the set occurrence.

Action: Review associated messages and determine if fixing should be performed.

070E SECOND OWNER RECORD RETRIEVED

Reason: When walking either next or prior, a second owner record occurrence was detected in the set occurrence. This message can indicate a set occurrence that points into another set occurrence.

Action: Review associated messages and determine if fixing should be performed.

071E SET LOOP DETECTED

Reason: When walking either next or prior in a set occurrence, the designated db-key of the displayed record pointed to a member record previously retrieved, indicating a set loop condition.

Action: Review associated messages and determine if fixing should be performed.

072E FRAGMENTED RECORD NOT PROPERLY ASSEMBLED

Reason: All fragments of the record could not be successfully retrieved because of a bad SR4 chain pointer. The record is either variable length or compressed. (A compressed fixed length record is stored on the database as a variable length record. See message 080E.) The Record Descriptor Word (RDW) length could also erroneously indicate that there should be more fragments for the record while the last SR4 chain pointer points back to the root of the fragmented record indicating that there are no more fragments. Only the control portion of the record data is displayed.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE to display the page where the designated record resides and patch the bad page.

077E CA-IDMS/DB Audit TERMINATED

Reason: Processing was terminated because of a previous error.

Action: Review all displayed error messages.

078E AREA/RECORD/SET DESCRIPTOR BLOCK NOT FOUND IN SUBSCHEMA

Reason: The extract file used by the current fix/report phase was created by another CA-IDMS/DB Audit job. The current subschema being used is not the same subschema as that used by the audit phase even though the subschema names match.

Action: Check that the correct load/core image library is being used. Resubmit the job. If the problem recurs, save all output listings and contact Computer Associates Product Support for assistance.

079E INTERNAL PROCESSING ERROR

Reason: An irrecoverable internal error was detected by CA-IDMS/DB Audit.

Action: If CURRENCY/AREA-SWEEP is indicated, a page header may be invalid. Rerun, specifying AUDIT=PAGES. Otherwise, save all output listings and contact Computer Associates Product Support for assistance.

080E COMPRESSED RECORD NOT PROPERLY DECOMPRESSED

Reason: The displayed record is a compressed record that could not be successfully decompressed because of errors detected within the compressed portion of the record. Improper decompression can cause a fragmented record to be improperly assembled. See message 072E.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE to display the page where the designated record resides and patch the bad page.

081E NOT ENOUGH STORAGE FOR LOADING SUBSCHEMA/DMCL

Reason: The amount of available storage in the region or partition was not large enough for CA-IDMS/DB Audit to load the subschema or the DMCL named in the subschema, or both, into virtual storage.

Action: See the “Operations” chapter for details about CA-IDMS/DB Audit storage requirements. Increase the GETMAIN/GETVIS area by changing the REGION/SIZE parameter on the EXEC statement. Resubmit the job.

082E SUBSCHEMA/DMCL NOT VALID

Reason: The subschema and/or the DMCL named in the subschema resides in the load/core image libraries allocated to the job. CA-IDMS/DB Audit has determined, however, that the subschema is not a valid subschema and/or the DMCL named in the subschema is not a valid DMCL.

Action: Check the spelling of the subschema named in the PROCESS statement. Check that the correct load/core image libraries were allocated. Resubmit the job.

084E AUDIT,DISCONNECT=EXPLICIT SPECIFIED BUT NO MEMBER RECORD IS M/A OR DISCONNECT

Reason: DISCONNECT=EXPLICIT was specified on the AUDIT statement indicating that CA-IDMS/DB Audit should audit for set disconnections of mandatory automatic or DISCONNECT member records. There are no records, however, that are to be audited that are mandatory automatic or specified as DISCONNECT.

Action: Delete DISCONNECT=EXPLICIT from the AUDIT statement, or specify specific records or sets as DISCONNECT, and resubmit the job.

085E DISCONNECT SPECIFIED FOR RECORD/SET BUT DISCONNECT=NONE WAS SPECIFIED ON AUDIT STATEMENT

Reason: DISCONNECT=NONE was specified in the AUDIT statement, indicating that CA-IDMS/DB Audit should not audit for set disconnections. However, at least one RECORD or SET statement specified DISCONNECT, and CA-IDMS/DB Audit terminated processing.

Action: Specify DISCONNECT=EXPLICIT in the AUDIT statement. Delete DISCONNECT from all RECORD and SET statements. Resubmit the job.

086E READYMODE=RETRIEVAL AND FIX=UPDATE WERE BOTH SPECIFIED BUT ARE INCOMPATIBLE

Reason: Retrieval mode and updating the database are mutually exclusive.

Action: If you want to run in retrieval mode, change to FIX=SIMULATE. If you want to actually update the database, change to READYMODE=UPDATE. Resubmit the job.

087E SR8 AUDITING TERMINATED

Reason: Date and time that SR8 auditing was terminated because of a processing error.

Action: Review associated messages to determine the reason for termination.

088E RECORD NOT FOUND - RECORD NAME IS record-name

Reason: The record indicated by a db-key contained in an SR8 entry could not be found.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to fix the SR8 record.

089E RECORD TYPE INCORRECT - RECORD NAME IS record-name

Reason: The record indicated by a db-key contained in an SR8 entry was found, but it is not the correct record type for this integrated index set.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the record resides and patch the bad page.

090E RECORD NOT WITHIN PAGE RANGE - RECORD NAME IS record-name

Reason: The record indicated by a db-key in an SR8 entry was found, but its page range is not within the page range indicated for that record type.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the record resides and patch the bad page.

091E RECORD LOCATED IN A NON - READIED AREA - RECORD NAME IS record-name

Reason: The record indicated by a db-key in an SR8 entry resides in an area that was not readied.

Action: Remove the NOACCESS parameter for the area where the record resides. Resubmit the job.

092E INDEX ORPHAN NOT FIXED DUE TO AN INVALID NEXT POINTER

Reason: The orphan count in the SR8 record was not fixed because an invalid next pointer was detected when trying to decrease orphan counts. CA-IDMS/DB Audit should have detected and already fixed this pointer.

Action: Make sure the correct audit extract file is being used. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to correct the invalid next pointer in the SR8 record and fix the orphan counts.

093E INVALID RETURN CODE RECEIVED FROM PROGRAM program-name RETURN CODE IS return-code

Reason: An error occurred during the execution of the indicated program.

Action: Contact Computer Associates Product Support.

094E CURRENT SUBSCHEMA DOES NOT MATCH DATABASE

Reason: The subschema specified in the PROCESS statement does not match the database. The database contains an SR7 record for a system owned integrated index set, but the subschema does not contain a definition for that set, nor does the subschema indicate that it is a partial description of the database. The subschema for CA-IDMS/DB Audit must be created from a schema that has a complete description of the database.

Action: Correct the subschema and resubmit the job.

095E AN INTERNAL SORT IN PROGRAM program-name DETECTED A RETURN CODE OF sort-return-code

Reason: The sort program was unable to complete a sort.

Action: Examine the related messages from your sort utility. See the list of return codes associated with the sort utility for appropriate action.

096E UNSUCCESSFUL READ OF file-name FILE

Reason: CA-IDMS/DB Audit was unable to read the indicated file.

Action: See message 050I.

097E UNSUCCESSFUL WRITE TO file-name FILE

Reason: CA-IDMS/DB Audit was unable to write to the indicated file.

Action: See message 050I.

098E UNSUCCESSFUL CLOSE OF file-name FILE

Reason: CA-IDMS/DB Audit was unable to close the indicated file.

Action: See message 050I.

099E SET set-name FOUND ON INDEX DBKEY WORK FILE BUT NOT FOUND IN SUBSCHEMA

Reason: DISP=MOD on the index db-key work file (instead of DISP=OLD, DISP=SHR, or DISP=NEW) causes the output to be added to the existing file. This problem can occur if CA-IDMS/DB Audit is run with two different subschemas.

Action: Run CA-IDMS/DB Audit with DISP=OLD, DISP=SHR, or DISP=NEW.

100E AUDIT,INDEXES/USERSETS SPECIFIED BUT SR7 RECORD SET IS NOACCESS

Reason: The integrated index set cannot be audited because the set or a record of the set was specified NOACCESS in a RECORD or SET statement.

Action: Delete INDEXES from the AUDIT parameter statement or delete the RECORD or SET statement that specifies NOACCESS. Resubmit the job.

101E AUDIT,INDEXES/USERSETS SPECIFIED BUT SR7 RECORD IS NOSWEEP

Reason: The integrated index set cannot be audited because the SR7 record is designated NOSWEEP in a RECORD parameter statement.

Action: Delete INDEXES from the AUDIT parameter statement or delete the RECORD statement that specifies NOSWEEP for the SR7 record. Resubmit the job.

102E SR8 UPPER LEVEL POINTER IS NOT HIGH VALUES

Reason: In a top level SR8 record, the upper level pointer is not null (High-Values).

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to patch the upper level pointer.

103E INVALID GSSCALL RETURN CODE DETECTED WHEN CALLING PROGRAM program-name. RETURN CODE IS nnnn.

Reason: A module call processor error occurred.

Possible return codes are:

- **4** Improper parameter list
- **8** Module not found in load/core image library
- **12** Not enough storage for internal table or module (DOS only)
- **16** GSSANKR could not be loaded.

Action: Contact Computer Associates Product Support.

104E INTGR INDEX ORPHAN COUNT NOT DECREMENTED. INVALID RECORD TYPE ENCOUNTERED.

Reason: INDEXORPHANS was specified in the FIX statement. Because an invalid record type was encountered, the integrated index orphan count was not decreased.

Action: Correct the invalid record. You can use the CA-IDMS utility IDMSBCF, FIX PAGE function to decrease the orphan count.

105E INTGR INDEX ORPHAN COUNT NOT DECREMENTED. MEMBER RECORD UPPER LEV DBKEY POINTS OUT OF SET.

Reason: INDEXORPHANS was specified in the FIX statement. Because of an invalid pointer pointing out of the index set, orphan counts could not be completely decreased.

Action: Fix the erroneous pointer by using the CA-IDMS utility IDMSBCF, FIX PAGE function. You can use IDMSBCF, FIX PAGE to decrease the orphan count.

106E DISK/CORE OPTIONS ARE MUTUALLY EXCLUSIVE.

Reason: Both DISK and CORE were entered in the AUDIT parameter statement.

Action: Remove either DISK or CORE from the AUDIT statement and resubmit the job. If DISK is specified, be sure that two VSAM files are allocated.

107E GSIDCOP UNABLE TO DECOMPRESS SR8 RECORD return code

Reason: An SR8 record has been encountered where the decompressed record size exceeds Max-SR8-size as calculated from the subschema. This calculation is based upon the INDEX BLOCK CONTAINS *n* KEYS in the schema description of the integrated index set.

Action: Rerun using a subschema created from a schema having a larger value for INDEX BLOCK CONTAINS *n* KEYS.

108E MUST USE THE DISK DBKEYTBL OPTION IF SETLIMIT OVER 300,000 - YOU ASKED FOR number-of-bytes BYTES

Reason: Setlimit is an important factor in determining the size of the db-key table. In this case, setlimit is so large that the db-key table cannot reside in core.

Action: Specify DISK on the AUDIT statement and resubmit the job. If DISK is specified, be sure that two VSAM files are allocated.

109E ONLY 1 SWEEP/NOSWEEP/NOACCESS KEYWORD CAN BE ENTERED PER AREA STATEMENT

Reason: Only one of these keywords is allowed per AREA statement.

Action: Determine which keyword is desired and resubmit job.

110E A1-A7 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or change the AREA statement to allow access, then resubmit the job.

111E A1-A8 NOACCESS AREA AND AUDIT SET CONFLICT

Reason: A set that is to be audited resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or change the AREA statement to allow access, then resubmit the job.

112E ONE SWEEP/NOSWEEP/NOACCESS SUBPARAMETER MUST BE ENTERED FOR EACH AREA STATEMENT

Reason: One of these subparameters must be specified on with the AREA statement.

Action: Specify subparameter for area and resubmit the job.

113E R4-R3 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or change the AREA statement to allow access, then resubmit the job.

114E R4-R8 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or change the AREA statement to allow access, then resubmit the job.

115E R4-R10 NOACCESS AREA AND AUDIT SET CONFLICT

Reason: A set that is to be audited reside in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or change the AREA statement to allow access, then resubmit the job.

116E R4-R11 NOACCESS AREA AND AUDIT SET CONFLICT

Reason: A set that is to be audited reside in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or change the AREA statement to allow access, then resubmit the job.

117E S3-S2 NOACCESS AREA AND AUDIT SET CONFLICT

Reason: A set that is to be audited reside in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or change the AREA statement to allow access, then resubmit the job.

118E S3-S10 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or change the AREA statement to allow access, then resubmit the job.

119E S4-S2 NOACCESS AREA AND AUDIT SET CONFLICT

Reason: A set that is to be audited resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or change the AREA statement to allow access, then resubmit the job.

120E R11-R4 AUDIT SET AND NOACCESS AREA CONFLICT

Reason: A set that is to be audited resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or changing the AREA statement to allow access, then resubmit the job.

121E S4-S10 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or changing the AREA statement to allow access, then resubmit the job.

122E INDEX ORPHAN FIXING IS NOT AVAILABLE WITH QUICKCHECK

Reason: Index orphan fixing was specified on the FIX statement in a QuickCheck type audit run.

Action: Remove INDEX ORPHAN FIX parameter or change audit type to Standard and resubmit the job.

123E ONLY 1 SWEEP/NOSWEEP/NOACCESS KEYWORD CAN BE ENTERED PER RECORD STATEMENT

Reason: Only one of these keywords is allowed per RECORD statement.

Action: Determine which keyword is desired and resubmit job.

124E R1-R10 NOACCESS RECORD AND SET AUDIT CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

125E R1-R11 NOACCESS RECORD AND SET AUDIT CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

126E S7-S2 NOACCESS RECORD AND SET AUDIT CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

127E S8-S2 NOACCESS RECORD AND SET AUDIT CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

128E ONE SWEEP/NOSWEEP/NOACCESS SUBPARAMETER MUST BE ENTERED FOR EACH RECORD STATEMENT

Reason: One of these subparameters must be specified on the RECORD statement

Action: Specify subparameter for record and resubmit the job.

129E R3-R4 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or changing the AREA statement to allow access, then resubmit the job.

130E A7-A1 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or changing the AREA statement to allow access, then resubmit the job.

131E S10-S3 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or changing the AREA statement to allow access, then resubmit the job.

132E S10-S4 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or changing the AREA statement to allow access, then resubmit the job.

133E R8-R4 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: A record set to SWEEP resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the RECORD statement or changing the AREA statement to allow access, then resubmit the job.

134E S2-S3 AUDIT SET AND NOACCESS AREA CONFLICT

Reason: A set that is to be audited resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or changing the AREA statement to allow access, then resubmit the job.

135E S2-S4 AUDIT SET THE NOACCESS AREA CONFLICT

Reason: A set that is to be audited resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or changing the AREA statement to allow access, then resubmit the job.

136E S2-S7 AUDIT SET AND NOACCESS AREA CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

137E S2-S8 AUDIT SET AND NOACCESS RECORD CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

138E R10-R4 AUDIT SET AND NOACCESS RECORD CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

139E R10-R4 AUDIT SET AND NOACCESS AREA CONFLICT

Reason: A set that is to be audited resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or changing the AREA statement to allow access, then resubmit the job.

140E A8-A1 AUDIT SET AND NOACCESS RECORD CONFLICT

Reason: A set that is to be audited resides in an area that is set to NOACCESS.

Action: Resolve the conflict by either deleting the SET statement or changing the AREA statement to allow access, then resubmit the job.

141E R11-R1 AUDIT SET AND NOACCESS RECORD CONFLICT

Reason: A record set to NOACCESS is a member of a set that is to be audited.

Action: Resolve the conflict by either deleting the SET statement or changing the RECORD statement to allow access, then resubmit the job.

142E ONE AUDIT/NOAUDIT SUBPARAMETER MUST BE ENTERED FOR EACH SET STATEMENT

Reason: One of these subparameters must be specified on each SET statement.

Action: Specify subparameter for set and resubmit the job.

143E INTEGRITY ERROR ENCOUNTERED PROCESSING IIX SET index-set-name

Reason: SR7 or SR8 record was found to have inconsistent next, prior, or owner pointers. This error is reported during AUDIT INDEX processing when AUDIT USERSETS was not also specific.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the SR8 resides to determine problem with the index set pointers, and to patch the bad page.

144E INDEX AUDITING IS NOT AVAILABLE WITH QUICKCHECK

Reason: Audit subparameter is specified with QuickCheck audit.

Action: Index integrity checking is not available with QuickCheck type audit.

145E WALKNEXT ORPHAN DETECTION IS AUTOMATICALLY INCLUDED WITH QUICKCHECK

Reason: WALKNEXT subparameter is specified with QuickCheck type audit.

Action: USERSET orphan detection is incorporated with QuickCheck type audit.

146E QUICKCHECK AUDIT PREVIOUSLY SPECIFIED - FIX PHASE MUST ALSO RUN WITH QUICKCHECK

Reason: QuickCheck type audit created the extract file but Standard type audit was specified in subsequent fix run.

Action: Change audit type in fix run to Standard or rerun audit type Standard to create new extract file.

147E STANDARD AUDIT PREVIOUSLY SPECIFIED - FIX PHASE MUST ALSO RUN WITH STANDARD

Reason: Standard type audit created the extract file, but QuickCheck type audit was specified in subsequent fix run.

Action: Change audit type in fix run to QuickCheck or rerun with audit type QuickCheck to create new extract file.

148E QUICKCHECK USERSETS/CALCSETS NOT AVAILABLE WITH BEGIN/END PAGE RANGE FOR AREA area-name

Reason: BEGIN/END subparameter specified with AREA statement and QuickCheck audit requires a sweep of all records in the area.

Action: If partial area audit is preferable, then change audit type to Standard or audit all records in the area as required with QuickCheck type audit.

149E THE FOLLOWING INTEGRATED INDEX SETS MUST BE PROCESSED DUE TO AUDIT INDEX SET index-set-name

Reason: If any index set is to be audited, all index sets in that page range must be audited.

Action: Change parameters so that all index sets that follow are audited to ensure correct index processing.

150E SUBSCHEMA CONTAINS ONE OR MORE NATIVE VSAM FILES

Reason: Native VSAM files are not allowed by CA-IDMS/DB Audit.

Action: VSAM file must be removed from subschema in order for CA-IDMS/DB Audit to be run.

164E SWEEP SR7 RECORD UNSUPPORTED

Reason: SWEEP record type SR7 specified on RECORD statement.

Action: SWEEP SR7 cannot be used to initiate audit of index set.

167E INTG INDEX ORPHAN COUNT NOT DECREMENTED. MEMBER RECORD UPPER LEV DBKEY WAS INVALID

Reason: Index orphan has an invalid upper level index record pointer.

Action: Index orphan is connected to correct upper level index record but, the invalid pointer value caused the counter not to be decremented.

168E QUICKCHECK ERROR NOT FOUND WALKING FROM THE FOLLOWING DBKEYS: db-key(s)

Reason: Error found during QuickCheck sort and match, but no evidence of error could be found when SETWALK process was invoked to determine the cause of the error.

Action: Rerun audit with process READYMODE set to UPDATE to assure no database update will occur during the audit run. If problem still exists, notify Computer Associates Product Support.

169E QUICKCHECK POINTER EXTRACT MISMATCH WITH THE FOLLOWING EXTRACT: extract(s)

Reason: Pointer error(s) were found while QuickCheck was processing an integrated index set.

Action: QuickCheck is not able to determine correct index set sequence. Run Standard audit with AUDIT INDEX option set on.

170E LOAD OF GSDTPARM OR GSDBLDP FAILED

Reason: An internal error has occurred.

Action: Contact Computer Associates Product Support.

171E INVALID USER WORK AREA SIZE PASSED

Reason: The user work area size must be a value 64 to 256 inclusive.

Action: Correct the user work area size and resubmit the job.

172E SUBSCHEMA AREA NOT FOUND IN THE DMCL

Reason: All the areas in the subschema are not included in the DMCL.

Action: Use a DMCL that contains all the areas in the subschema.

174E SUBSCHEMA CONTAINS AREAS WITH MIXED PAGE GROUPS OR MIXED RADIX - NOT SUPPORTED

Reason: Subschema contains areas with mixed page groups or mixed radix - not supported.

Action: CA-IDMS/DB Audit cannot be run against a subschema that contains mixed page groups or mixed radix.

905E GSSFILE RETURNED AN ERROR DURING function WITH FILE ddname, RETURN CODES are N1, N2, N3, N4

Reason: The indicated file function could not be performed with the specified file.

Action: See Figure 6.2 for an explanation and appropriate action for the return codes indicated.

6.3 Informative Messages

Informative messages are suffixed by an *nnn*I message number and include possible causes for the message and any actions to be taken by the user.

001I INPUT PARAMETER STATEMENT

Reason: A CA-IDMS/DB Audit input parameter statement has been read. The entire 80 byte image is displayed.

Action: None.

002I PARAMETER STATEMENT CONTINUED

Reason: The parameter statement listed in message 001I contains a comma after the last keyword phrase. No processing is performed until a complete parameter statement is read.

Action: None.

003I COMMENT STATEMENT IGNORED

Reason: The parameter statement listed in message 001I contains an asterisk (*) or a dollar sign ($) in position 1. The statement is not processed by CA-IDMS/DB Audit.

Action: None.

004I NO PARAMETER ERRORS DETECTED

Reason: A complete parameter statement was processed and no errors were detected.

Action: None.

005I PARAMETER STATEMENT NOT PROCESSED

Reason: The parameter statement listed in message 001I was not processed because of parameter errors.

Action: Review the associated error messages. Correct the errors and resubmit job.

006I INPUT PARAMETER STATEMENT FLUSHED

Reason: A previous error has made execution of CA-IDMS/DB Audit impossible; all remaining parameter statements are displayed but not processed.

Action: Review the associated error messages. Correct any errors and resubmit the job.

007I END-OF-FILE ON SYSIPT

Reason: The end of the input parameter file was reached. All parameter statements were read.

Action: None.

008I number-of-messages ERROR/WARNING LEVEL PARAMETER ERRORS DETECTED

Reason: The number of error or warning messages is shown. If any processing errors were detected, CA-IDMS/DB Audit terminates processing.

Action: Correct any errors and resubmit the job. Review all warning messages to determine if the requested processing options were in effect. If required, correct the parameter causing the warning message and resubmit the job.

009I END PARAMETER PROCESSING

Reason: All parameter statements were processed and no errors were detected.

Action: None.

010I INITIATING CA-IDMS/DB Audit--date time

Reason: CA-IDMS/DB Audit has successfully completed its parameter processing phase. CA-IDMS/DB Audit prepares to obtain the amount of storage required for processing and initiate CA-IDMS housekeeping functions at the designated date and time.

Action: None.

011I STORAGE-POOL SIZE=number-of-bytes BYTES

Reason: CA-IDMS/DB Audit displays the total amount of main storage obtained for its work areas (record/page bind areas, symbolic sort key areas, DBKEYTBL, and db-key save table). See the “Operations” chapter for more details about CA-IDMS/DB Audit storage requirements.

Action: None.

012I INITIATING RECORD/PAGE BINDS--number-of-bytes BYTES OF STORAG E ALLOCATED

Reason: CA-IDMS/DB Audit prepares to bind all records and pages that are to be audited, fixed, or reported. The amount of storage allocated from the storage pool for the BIND areas and symbolic sort key work areas is displayed.

Action: None.

013I INITIATING READIES

Reason: CA-IDMS/DB Audit prepares to ready all areas to be audited, fixed, or reported for EXCLUSIVE UPDATE or SHARED RETRIEVAL according to the READYMODE parameter of the PROCESS statement. The names of areas successfully readied follow this message.

Action: None.

014I INITIALIZING DBKEYTBL--number-of-bytes BYTES OF STORAGE ALLOCATED

Reason: CA-IDMS/DB Audit initializes the DBKEYTBL that is used for determining whether or not a set occurrence contains a loop. The amount of storage allocated from the core storage pool is displayed. If DISK is specified in the AUDIT statement, the number of bytes is 0 (zero). See the “Parameters” chapter for more information about the DBKEYTBL.

Action: None.

015I INITIALIZING DBKEY-SAVE TABLE--number-of-bytes BYTES OF STORAGE ALLOCATED

Reason: CA-IDMS/DB Audit initializes the db-key save table that is used for auditing a set containing prior pointers. The amount of storage allocated from the storage pool is displayed. If DISK is specified in the AUDIT statement, the number of bytes is 0 (zero). See Set Walk Processing in the “Concepts” chapter for more information about the db-key save table. This table is also used when processing integrated index sets.

Action: None.

016I INITIATING AUDIT/FIX/REPORT PHASE

Reason: Displays the date and time the designated phase began processing.

Action: None.

017I END AUDIT/FIX/REPORT PHASE, RC=nn

Reason: Displays the date and time the designated phase ended processing, and its return code.

Action: None.

018I BEGINNING SWEEP OF AREA area-name

Reason: Displays the date and time CA-IDMS/DB Audit began sweeping the designated area during the audit phase.

Action: None.

019I END OF AREA SWEEP

Reason: Displays the date and time CA-IDMS/DB Audit ended the sweep of the area designated in message 018I.

Action: None.

022I AUDIT/FIX PHASE STATISTICS

Reason: Associated audit and fix phase statistics follow this message.

Action: See the “System Output” chapter for a description of the statistics.

023I IDMS STATISTICS FOR AUDIT/FIX PHASE

Reason: Associated CA-IDMS and Computer Associates statistics follow this message. This message follows informative message 017I.

Action: None.

024I UTILIZING INPUT EXTRACT FILE CREATED

Reason: A FIX parameter statement was entered without an AUDIT parameter statement. CA-IDMS/DB Audit uses as input an extract file created on the designated date and time by a previous CA-IDMS/DB Audit job.

Action: None.

025I NO FIXABLE ERROR/LDEL RECORDS DETECTED IN AUDIT PHASE OR NO EXTRACTS FOUND FOR FIX REQUEST TYPE

Reason: A request to physically delete LDEL records was made, but no LDEL records were detected in the audit phase. A request to fix CALCSETS, USERSETS, and/or PAGES was made, but no physical integrity errors were detected during the audit phase.

Action: Review the FIX parameter statement to verify if the correct FIX request was made.

026I NO INTEGRITY ERRORS DETECTED DURING AUDIT PHASE

Reason: No integrity errors were detected during the audit phase. No Integrity Analysis Report will be produced, even though a REPORTS statement is present.

Action: None.

027I PROCESSING OPTIONS IN EFFECT

Reason: Following this message, CA-IDMS/DB Audit displays all processing options in effect for the current job.

Action: None.

028I END OF CA-IDMS/DB Audit

Reason: CA-IDMS/DB Audit successfully completed the requested processing phases (audit, fix, or reports) at the indicated date and time.

Action: None.

029I PAGE INTEGRITY ERRORS DETECTED IN AREA area-name

Reason: During the audit phase, physical integrity errors were detected in the designated area when auditing the PAGE header/footer.

Action: Review the associated error messages and determine if fixing should be performed.

030I FIXING PERFORMED ON PAGES IN AREA area-name

Reason: Fixing was requested. CA-IDMS/DB Audit has fixed the pages in the designated area as shown.

Action: Refer to FIX Report banner to determine if real or simulated fixing was done. If simulated fixing was done, review the fixes to determine if real fixes should be applied.

031I RECORD DBKEYS OUTSIDE SPECIFIED FROM/THRU PAGE RANGES

Reason: The db-keys of the displayed records do not fall within the from/thru page ranges specified. The records do not participate in a set occurrence found to contain other integrity errors.

Action: None. CA-IDMS/DB Audit cannot fix this type of error.

Note: This type of error does not affect the walking of a set occurrence. It can, however, cause CALC records not to be found.

032I RECORDS ARE FULLY DISCONNECTED FROM SET set-name

Reason: The auditT phase has determined that the designated record occurrences are fully disconnected from a set occurrence. Either DISCONNECT=ALL or DISCONNECT=EXPLICIT was specified.

Action: If the record(s) should be connected to a set, use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the designated record resides and patch the bad page.

033I POINTER ERRORS DETECTED FOR RECORDS IN SET set-name

Reason: The records displayed do not participate in a set occurrence found to contain other integrity errors. A partial area audit was performed and the record occurrence is a member of a set occurrence whose owner record was never retrieved through the area sweep. A full area audit was performed and the record occurrence is an orphan of a set occurrence and WALKNEXT was not requested.

Action: Execute an audit with a full area sweep where the owner record of the indicated set resides. Execute an audit requesting WALKNEXT processing. In order for processing to proceed quickly, all areas, records, and sets that do not participate in the error set can be designated as NOACCESS in the appropriate AREA, RECORD, or SET parameter statements.

034I INTEGRITY ERRORS DETECTED IN SET set-name

Reason: The audit phase has determined that the designated set contains the following integrity errors. Information about the owner record of the set follows this message.

Action: Review the associated messages and determine if fixing should be performed.

035I FIXING PERFORMED ON SET set-name

Reason: When fixing was requested, CA-IDMS/DB Audit fixed the designated set as shown. Information about the owner record of the set follows this message.

Action: Refer to the FIX Report banner to check if real or simulated fixing was performed. If simulated fixing was performed, review the fixes to determine if real fixes should be applied.

036I RECORDS DETECTED WALKING NEXT FROM OWNER

Reason: Information about every record occurrence detected while walking next from the owner record of the set follows this message.

Action: Review the associated messages.

037I RECORDS DETECTED WALKING PRIOR FROM OWNER

Reason: Information about every record occurrence detected while walking prior from the owner record of the set follows this message.

Action: Review the associated messages and determine if fixing should be performed.

038I RECORDS DETECTED WALKING NEXT/PRIOR FROM record-db-key

Reason: This message appears only for a set containing both next and prior pointers. While walking next, the designated record's prior db-key did not point back to the prior record in the set. While walking prior, the designated record's next db-key did not point back to the next record in the set. Once the chain containing the record with the mismatched next/prior db-key is walked, CA-IDMS/DB Audit begins walking the chain from the record in the opposite direction. See Set Walk Processing in the “Concepts” chapter for more information.

Action: Review the associated messages and determine if fixing should be performed.

039I NEXT/PRIOR DBKEY POINTS TO PREVIOUSLY AUDITED RECORD IN SET

Reason: While walking a set occurrence containing an integrity error, CA-IDMS/DB Audit detected that the record's designated db-key pointed to a member record that was already audited.

Action: Review the associated messages and determine if fixing should be performed.

040I RECORD IS LOGICALLY DELETED

Reason: Information about a record occurrence is displayed. The record is logically deleted, however, and does not have any record data associated with it.

Action: None.

041I RECORDS ARE ORPHANS OF SET record-name(s)

Reason: The record occurrences displayed after this message were detected as orphan records via walknext processing.

Action: Review the associated messages and determine if fixing should be performed. If fixing is performed, the displayed orphan records will be added back into the set.

042I ORPHANS POINT INTO SET set-name

Reason: CA-IDMS/DB Audit has determined that the designated set contains no physical integrity errors. During walknext processing, however, orphan records were found to be associated with the set occurrence.

Action: Review the associated messages and determine if fixing should be performed. If fixing is performed, the orphan records will be added back into the set.

043I ORPHANS NOT CONNECTED TO ANY OWNER RECORD

Reason: The records following this message could not be associated with any owner record occurrences for the indicated set type.

Action: None. CA-IDMS/DB Audit cannot fix this type of error.

044I LDEL RECORDS PHYSICALLY DELETED FROM SET set-name

Reason: When fixing was requested, CA-IDMS/DB Audit successfully deleted the logically deleted records from the indicated set occurrence. The deleted records are listed following this message.

Action: Refer to the FIX Report banner to check if real or simulated fixing was performed. If simulated fixing was performed, review the fixes to determine if real fixes should be applied.

045I RECORD ERASED

Reason: When fixing was requested, CA-IDMS/DB Audit has determined that the displayed LDEL record was physically deleted from all sets where it was a member. CA-IDMS/DB Audit erases the LDEL record occurrence.

Action: Refer to the FIX Report banner to check if real or simulated fixing was performed. If simulated fixing was performed, review the fixes to determine if real fixes should be applied.

048I FRAGMENTED RECORDS NOT PROPERLY ASSEMBLED

Reason: This message is displayed only for records that do not participate in a set occurrence found to contain other integrity errors. Two conditions can cause this message:

1. The SR4 chain pointer for a fragmented record does not point correctly to all of the fragments of the record. The record is either a variable length record or a compressed fixed length record (which is stored on the database as a variable length record). The message can be caused by a compressed record not properly decompressed (see message 054I).
2. The last SR4 chain pointer points back to the root of the fragmented record while the Record Descriptor Word (RDW) length indicates that there should be more fragments.

Only the control portion of the record data is displayed.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page or pages where the designated records reside and to patch the bad page or pages.

049I INITIATING ACTUAL FIXING OF SETS/LDELS/PAGES/INDEX ORPHANS

Reason: CA-IDMS/DB Audit prepared for correction of either SETS (CALCSETS and/or USERSETS), LDELS, or PAGES at the designated date and time.

Action: None.

050I ERROR STATUS = error-code

Reason: The displayed non-zero error status was returned after executing a CA-IDMS DML command. This message follows the error message that describes the type of CA-IDMS function requested.

Action: Review the error status and take appropriate action. See Figure 6.1 at the end of this chapter for a description of error status codes not described in the *CA-IDMS CODES and MESSAGES manual*.

051I RECORD OBTAINED THROUGH AREA SWEEP record-name

Reason: This message follows any error messages that caused CA-IDMS/DB Audit to terminate processing on a specific set occurrence. The name of the last record retrieved through the area sweep is displayed.

Action: None.

052I hex-db-key decimal-db-key

Reason: The displayed db-key is associated with the previously displayed error message. Both hexadecimal and decimal forms are displayed.

Action: None.

053I OWNER RECORD OF SET record-name

Reason: CA-IDMS/DB Audit displays the record name of the owner of the set occurrence where CA-IDMS/DB Audit terminated processing.

Action: Review the associated informative and error messages.

054I COMPRESSED RECORDS NOT PROPERLY DECOMPRESSED

Reason: This message appears only if the records do not participate in a set occurrence found to contain other integrity errors. The records displayed following this message are compressed records that could not be successfully decompressed because of errors within the compressed portion of the records. A record that could not be properly decompressed can cause CA-IDMS/DB Audit to report that fragments could not be properly assembled. See message 048I.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page or pages where the designated records reside and to patch the bad page or pages.

055I DATABASE SUCCESSFULLY UPDATED

Reason: CA-IDMS/DB Audit has successfully completed fixing of SETS, LDELS, or PAGES displayed in message 049I. A COMMIT DML has been successfully issued and the database buffers have been cleared.

Action: None.

056I number-of-records RECORD(S) NOT DISPLAYED

Reason: SETMBRS=ERRSONLY was specified in the REPORTS statement. Therefore, the display shows only records containing errors or records immediately preceding or following them. The message indicates the number of records that are not displayed for this set.

Action: None.

057I INITIATING SR8 AUDITING

Reason: Displays the date and time CA-IDMS/DB Audit began integrated index processing.

Action: None.

058I END SR8 AUDITING

Reason: Displays the date and time CA-IDMS/DB Audit ended integrated index processing.

Action: None.

059I INTEGRATED INDEX ORPHAN RECORDS DETECTED

Reason: The records displayed after this message are orphan records of the indicated integrated index set. The upper level pointer does not point to the correct SR8 record in the level above.

Action: Review the associated messages and determine if fixing should be performed. If fixing is performed, for each orphan record listed, the upper db-key is modified to point to the appropriate record in the next higher level. Also, the orphan counts are decreased.

060I INTEGRATED INDEX RECORDS DISCONNECTED FROM SET set-name

Reason: The records displayed after this message are disconnected from the indicated integrated index set. Db-keys obtained during sweep processing and index processing are not paired.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to correct the situation.

061I INDEX ENTRY DBKEY ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET index-set-name

Reason: When CA-IDMS/DB Audit looked for a record indicated by the db-key pointers in an integrated index set, it either did not find the record or it found errors within the record.

Action: Examine the associated messages to determine the specific nature of the error.

062I SYMBOLIC KEY ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET index-set-name

Reason: In a set sorted by symbolic key, an SR8 record points to a member record, but the symbolic key listed in the SR8 record does not match the symbolic key of the member record.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the designated record resides and patch the bad page.

063I SORT KEY SEQUENCE ERRORS DETECTED IN INTEGRATED INDEX RECORDS CONTAINED IN SET index-set-name

Reason: In an upper level SR8 entry, the sort key is in error. In a set sorted by db-key, the upper level SR8 db-key does not match the last db-key in the record below. In a set sorted by symbolic key, an upper level symbolic key does not match the last symbolic key in the record below. In a lower level SR8, the sequence of the table within the SR8 record is in error.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the designated record resides and patch the bad page.

064I SORT SEQUENCE ORDER IS ascending/descending

Reason: The sequence order of a sorted set.

Action: None.

065I INITIATING SR8 DATABASE VALIDATION date time

Reason: Date and time integrated index disconnect processing began.

Action: None.

066I END SR8 DATABASE VALIDATION date time

Reason: Date and time integrated index disconnect processing ended.

Action: None.

067I INTEGRATED INDEX ORPHANS FIXED

Reason: The upper level db-keys of the listed records were corrected.

Action: The report banner indicates whether real or simulated fixing was performed. If simulated fixing was performed, review the changes to determine if real fixing should be performed.

068I LEVEL SEQUENCE ERRORS DETECTED IN INTEGRATED INDEXES

Reason: The level number in an SR8 record does not match the number of the level where the record resides.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the designated record resides and patch the record.

069I SORT FLAG INCONSISTENT WITH SYMBOLIC SORT FLAG IN INTEGRATED INDEX RECORD record-name

Reason: In an integrated index record, the symbolic flag is on even though the sorted set flag is off.

Action: Use the CA-IDMS utility IDMSBCF, FIX PAGE function to display the page where the designated record resides and patch the record.

070I OWNER DBKEY db-key

Reason: The displayed owner db-key is associated with previously displayed error message.

Action: None.

071I IIX SET index-set-name

Reason: The displayed index set name is associated with previously displayed error message.

Action: None.

072I SORT RECORD TYPE=sort-record-type SET=set-name POINTERS=pointer-type

Reason: Indicates whether sort record was created from an actual database record (type=sweep) or a generated match record (type=dummy), shows the name of the set (for IIX sets the name cannot be determined), and shows what pointers the set has (NEXT, NEXT/PRIOR, NEXT/OWNER, NEXT/PRIOR/OWNER).

Action: When SET=UNKNOWN-IIX-SET, run AUDITTYPE=STANDARD.

073I OWNER db-key CURRENT db-key NEXT db-key PRIOR db-key

Reason: Displays the db-keys in the actual sort record.

Action: None.

074I INTEGRATED INDEX INTEGRITY ERRORS DETECTED DURING AUDIT PHASE - SEE RUNTIME STATUS ON AUDIT REPORT

Reason: Indicates errors were found while QuickCheck was processing an integrated index set.

Action: Refer to Audit Report for details of error. Run AUDITTYPE=STANDARD to correct the error.

075I IDMS DML ERROR STATUS RETURNED WHILE EXECUTING A DML VERB - SEE RUNTIME STATUS ON AUDIT REPORT

Reason: An earlier step encountered a DML error.

Action: Refer to Audit Report for details of error.

076I ORPHANS RANDOMIZE INTO SET...CALC

Reason: A record having a location node of CALC was not a member of the CALC set. It's CALC key indicates the CALC set owner to which it should be connected.

Action: Review the associated messages and determine if fixing should be performed. If fixing is performed, the orphan records will be added back into the set.

6.4 Warning Messages

Warning messages are suffixed by a *nnn*W message number. They give information about parameters that may lead to unexpected results.

001W NOACCESS/NOSWEEP SPECIFIED FOR area-name/record-name

Reason: Both NOACCESS and NOSWEEP were requested for the designated area or record. NOACCESS takes precedence and NOSWEEP is ignored.

Action: If NOSWEEP is required, review all input parameter statements (listed on the Audit Report) to determine which statement specified NOSWEEP, or caused NOACCESS. If necessary, correct the parameter statements and resubmit the job.

002W ASSUME ALL CALC RECORDS COPIED INTO SUBSCHEMA

Reason: The CALCSETS operand was specified in the AUDIT statement. CA-IDMS/DB Audit cannot ensure that all CALC records in all areas to be swept were copied into the subschema being used. A successful audit requires (and CA-IDMS/DB Audit assumes) that they were copied.

Action: None.

003W ASSUME EXTRACT FILE BEING INPUT FROM PREVIOUS AUDIT

Reason: A FIX parameter was entered without an AUDIT parameter statement. CA-IDMS/DB Audit assumes an extract file from a previous CA-IDMS/DB Audit job is being input. CA-IDMS/DB Audit will open the extract file for input.

Action: If CA-IDMS/DB Audit terminates abnormally immediately after this message is issued, an input extract file does not exist. Supply an AUDIT parameter statement or supply the correct extract file. Resubmit the job.

004W PARTIAL AREA AUDIT PERFORMED - AREA SWEEP IMPLIED

Reason: CA-IDMS/DB Audit has performed a partial area audit for the area listing in the preceding message 001W.

Action: None.

005W ALL RECORDS IN AREA ARE NOACCESS/NOSWEEP

Reason: An area that was not designated as NOACCESS or NOSWEEP contains records that were designated as NOACCESS or NOSWEEP by RECORD or SET parameter statements. The area will not be swept.

Action: None.

006W ALL MEMBER RECORDS FOR SET IN ERROR NOT RETRIEVED

Reason: The db-key table filled when creating extracts for a set occurrence containing integrity errors. Information about the specific set occurrence precedes this message. Not all records for the set in error were retrieved. Fixing is not allowed.

Action: Review the associated informative and error messages.

007W LDEL AN OWNER OF NON-EMPTY SET

Reason: The displayed logically deleted record (LDEL) is an owner of a non-empty set occurrence. CA-IDMS/DB Audit did not delete the record. The set occurrence owned by the LDEL is displayed elsewhere on the Before Images Report.

Action: You can perform the following actions in sequence to erase the member records owned by the LDEL and then erase the LDEL:

1. Store a fictional record of the same type as the LDEL in error.
2. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to connect the fictional record to the first member record owned by the LDEL record, and the last member record owned by the LDEL if there are prior pointers, in the non-empty set.
3. Use the CA-IDMS utility IDMSBCF, FIX PAGE function to change the next and prior pointers (for the non-empty set) in the LDEL, so that it points back to itself (to indicate an empty set).
4. Erase the fictional record with all of its member records.
5. Rerun the fix phase of CA-IDMS/DB Audit to erase the LDEL.

008W AUDIT/FIX/AREA/RECORD/SET PARAMETER STATEMENTS IGNORED SINCE REPORTS = ONLY

Reason: In this run, CA-IDMS/DB Audit did not perform an audit or fix any errors, because a REPORTS=ONLY statement was also entered. The statement indicates that only the report phase is to be performed.

Action: None.

009W INTEGRATED INDEX SET NOT INCLUDED IN READIED AREA. SET NAME IS index-set-name

Reason: The database contains an SR7 or a system-owned integrated index set. The subschema does not contain a description for that set, but the subschema is a partial definition of the database in regard to system owned integrated index sets.

Action: None.

010W THIS UPPER LEVEL POINTER WAS CORRECTED PRIOR TO FIX PROCESSING

Reason: An upper level pointer denoting an index orphan was found to be in error during the audit phase and was fixed before the fix phase was run.

Action: Ensure that the correct version of extract files produced during the audit phase is being used.

011W NO DISCONNECT AUDITING IS PERFORMED FOR INTEGRATED INDEXES

Reason: The SETLIMIT value was exceeded for an integrated index set, as reported in the previous message, 054E. CA-IDMS/DB Audit bypassed disconnect auditing on all integrated index sets.

Action: Choose an action appropriate for the conditions in your database:

1. Investigate the other errors that were found in the CA-IDMS/DB Audit run.
2. Increase the SETLIMIT value and resubmit the job. You may also want to use the DISK option to increase the amount of storage available.
3. Keep the same SETLIMIT value and designate the indicated integrated index set NOACCESS. Resubmit the job.

012W A1-A5 NOACCESS AREA AND SWEEP AREA CONFLICT

Reason: Member records of owned set in sweep area reside in a NOACCESS area.

Action: Member records are not accessed.

013W A1-A15 NOACCESS AREA AND NOSWEEP RECORD CONFLICT

Reason: RECORD statement is set to NOSWEEP, which implies access to area, but area where records reside is set to NOACCESS.

Action: Records are not accessed.

014W A1-A6 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: Member records of owned set reside in a NOACCESS area.

Action: Records will not be accessed.

015W A1-A9 NOACCESS AREA AND AUDIT SET CONFLICT

Reason: Member records of owned set to be audited reside in an area that has been set to NOACCESS.

Action: Records are not accessed.

016W R4-R2 NOACCESS AREA AND NOSWEEP RECORD CONFLICT

Reason: RECORD statement is set to NOSWEEP, which implies access, but area where records reside is set to NOACCESS.

Action: Records are not accessed.

017W R4-R7 NOACCESS AREA AND SWEEP AREA CONFLICT

Reason: Member records of owned set reside in a NOACCESS area.

Action: Records are not accessed.

018W R4-R9 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: Member records of owned set reside in a NOACCESS area.

Action: Records are not accessed.

019W S6-S4 SWEEP AREA AND NOACCESS AREA CONFLICT

Reason: Member records of owned set in sweep area reside in a NOACCESS area.

Action: Member records are not accessed.

020W R7-R1 SWEEP AREA AND NOACCESS RECORD CONFLICT

Reason: Member records of owned set in sweep area reside in a NOACCESS area.

Action: Member records are not accessed.

021W R7-R4 SWEEP AREA AND NOACCESS AREA CONFLICT

Reason: Member records of owned set in sweep area reside in a NOACCESS area.

Action: Member records are not accessed.

022W R1-R7 NOACCESS RECORD AND SWEEP AREA CONFLICT

Reason: Member records of owned set in sweep area have been set to NOACCESS.

Action: Member records are not accessed.

023W R1-R8 NOACCESS RECORD AND SWEEP RECORD CONFLICT

Reason: Owner records that are flagged as sweep have set member records that are NOACCESS.

Action: Member records are not accessed.

024W R1-R9 NOACCESS RECORD AND SWEEP RECORD CONFLICT

Reason: Owner records that are flagged as sweep have set member records that are NOACCESS.

Action: Member records are not accessed.

025W S8-S10 NOACCESS RECORD AND SWEEP RECORD CONFLICT

Reason: Owner records that are flagged as sweep have set member records that are NOACCESS.

Action: Member records are not accessed.

026W R2-R4 NOSWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: Records set as NOSWEEP, which implies ACCESS, are flagged as NOACCESS due to area being NOACCESS.

Action: Member records are not accessed.

027W A5-A1 NOSWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: Records set as NOSWEEP, which implies ACCESS, are in area that is set to NOACCESS.

Action: Member records are not accessed.

028W S10-S8 NOSWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: Records set as NOSWEEP, which implies ACCESS, are in area that is set to NOACCESS.

Action: Member records are not accessed.

029W R8-R1 SWEEP RECORD AND NOACCESS RECORD CONFLICT

Reason: Owner records that are set to SWEEP have member records that are flagged as NOACCESS.

Action: Member records are not accessed.

030W R9-R1 SWEEP RECORD AND NOACCESS RECORD CONFLICT

Reason: Owner records that are set to SWEEP have member records in another area flagged as NOACCESS.

Action: Member records are not accessed.

031W R9-R4 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: Owner records that are set to SWEEP have member record in an area flagged as NOACCESS.

Action: Member records are not accessed.

032W A6-A1 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: Owner records that are set to SWEEP have member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

033W A9-A1 AUDIT SET AND NOACCESS AREA CONFLICT

Reason: Set that is flagged to be audited has member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

034W S4-S6 NOACCESS AREA AND SWEEP AREA CONFLICT

Reason: Set that is flagged to audit has member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

035W INCONSISTENT NEXT/PRIOR/OWNER POINTER TYPES IN MULTI-MEMBER SET

Reason: Pointer type flagged does not exist in all member records in set.

Action: Inconsistent pointer type is ignored in records where it exists.

036W A1-A10 NOACCESS AREA AND SWEEP AREA CONFLICT

Reason: Owner records in an area set to sweep have member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

037W A1-A11 NOACCESS AREA AND SWEEP RECORD CONFLICT

Reason: Owner records set to sweep have member records in another area flagged as NOACCESS.

Action: Member records are not accessed.

038W A1-A12 NOACCESS AREA AND AUDIT SET CONFLICT

Reason: Set that is flagged to audit has member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

039W A4-A1 SWEEP AREA AND NOACCESS AREA CONFLICT

Reason: Owner records in an area set to sweep have member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

040W A10-A1 SWEEP AREA AND NOACCESS AREA CONFLICT

Reason: Owner records in an area to sweep have member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

041W A11-A1 SWEEP RECORD AND NOACCESS AREA CONFLICT

Reason: Owner records that are set to sweep have member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

042W A12-A1 AUDIT SET AND NOACCESS AREA CONFLICT

Reason: Set that is flagged to be audited has member records in an area flagged as NOACCESS.

Action: Member records are not accessed.

043W record-name RECORD IS NOACCESS. THIS CANCELED CALCSET PROCESSING IN AREA area-name.

Reason: Record that is set to NOACCESS is also a CALC record.

Action: CALC set is not audited.

045W FSBUFFERS KEYWORD WAS BYPASSED - REFER TO SYSIDMS PREFETCH CAPABILITY

Reason: FSBUFFERS parameter is no longer valid.

Action: Use SYSIDMS PREFETCH as an alternative to FSBUFFERS.

```
       DBIO Major Error Status Codes
FS = FAST-SCAN
GP = FIND/OBTAIN PAGE
PR = PAGE RETURN
       DBIO Minor Error Status Codes
A1 = New page returned
A2 = Function not supported
A3 = Invalid page range
A4 = Invalid qualifier
A5 = Page not current of run unit
A6 = Area not bound
A7 = Page return not on
A8 = Page return already on
A9 = Fast scan not on
B1 = Fast scan already on
B2 = SR4 chain error
B3 = SR2/SR3 chain error
B4 = HDR/FTR page number mismatch
B5 = Error closing file
B6 = Record not logically deleted
B7 = SMP currency error with page return on
B8 = Bad 'line space used' length in footer
B9 = No space in page
C1 = Duplicate DBKEY
C2 = Invalid Store
C3 = Skipped index
C4 = Broken chain
C5 = IXDET member of multiple sets
C6 = Ignored page reserve
C7 = Invalid Length - discrepancy in RDW lengths encountered
     during attempted MODIFY RECORD
C8 = DMCL/SCHEMA CA-IDMS release mismatch
C9 = Decompress error
D1 = Invalid record length
D4 = Invalid page displacement
D5 = Invalid request
D7 = GSDTPARM cannot be loaded
D8 = Record not found in PSUB
D9 = Segment named in DBNAME table entry is mismatched with current DMCL
DA = DBNAME not valid or segment name cannot be used as DBNAME

 Minor return codes prefixed with an I indicate an internal error:
I1 = Invalid parameter list
I2 = Function not supported
I3 = File already open/close
I4 = Insufficient work space--record read is larger than subschema description
I5 = Invalid currency data
I6 = VSAM block error
I7 = Bad fast scan DMCL block
I8 = File not assigned
```

Figure 6.1: DBIO Error Codes

6.5 Return Codes

Two types of errors can be reported by the return codes of n1, n2, n3, and n4--non-VSAM file errors and VSAM file errors. The error is described by n2 and n4. For VSAM file errors, n4 is always equal to 28. The error is described by n1, n2, and n3. A general return code is given by n4 for both non-VSAM and VSAM errors. All return codes are decimal values.

n4	Reason	Action
4	End-of-file	Call Product Support.
8	Open error or file is not open	Look for JCL errors or for the use of improper files.
12	An I/O error has occurred	Find cause for I/O error.
16	Request not recognized	Call Product Support.
20	File was already opened	Call Product Support.
24	Parameter list error	Call Product Support.
28	VSAM error n1=R15 return code from VSAM n2=low order byte from R0 GENCB/MODCB type of error n3=VSAM feedback byte error in I/O request	Use n1, n2, and n3 to check for possible user errors. If there are no user errors, call Product Support.
32	Insufficient storage	Increase storage for job step.

36	SYNAD error occurred
	n1=byte 1 of DECB
	n2=byte 2 of DECB
	N3=byte 3 of DECB
		For BDAM files.

40	BPAM FIND error
	n1=R15
	n2=R0
		Use n1 and n2 (as described in Data Management Macro Instructions) to check for errors.

44	BPAM STOW error
	n1=R15
	n2=0
		Use n1 and n2 (as described in Data Management Macro Instructions) to check for errors.

n2	Reason	Action
0	n4=8, use of unopened file n4=24, parameter list error	Call Product Support. Call Product Support.
1	JCL/label overrode parm list	Remove DCB information from JCL and ensure that the correct files are referenced.

2	Parm list overrode JCL/label	Remove DCB information from JCL and ensure that the correct files are referenced.
3	Unrecognized request	Call Product Support.
4	MVS x13 ABEND trapped at open	Fix cause for x13 ABEND.
5	Tried to update seq. file	Call Product Support.
6	VSAM write at other than load	Call Product Support.
7	SOS table could not expand	Call Product Support.
8	MVS DCB open failed	Call Product Support.
9	SOS table buffer pointer lost	Call Product Support.
10	SOS table file CB not built	Call Product Support.
11	MVS DD statement Missing	Supply missing DD statement.
12	VSAM ACB open failed	Call Product Support.
13	Record format invalid	Call Product Support.
14	Macro format invalid	Call Product Support.
15	Record length not numeric	Call Product Support.
16	Record length too large	Call Product Support.
17	Block size not numeric	Call Product Support.
18	Block size too large	Call Product Support.
19	Invalid VSE sysname table	Assemble a valid sysname table.
20	VSE sysname table entry missing	Assemble a sysname table with an entry for the missing one.
21	VSE LU number too large	Use an LU number within range.
22	VSE sysname is not numeric or is misspelled	Correct to a valid sysname.
23	VSE sysname blank	Do not use blank sysname.
24	VSE LU not assigned	Call Product Support.
25	VSE DTF prototype missing	Call Product Support.
26	VSE logic module missing	Generate missing logic module.
27	VSE CCW mismatch	Call Product Support.
28	File is not a PDS	Allocate file to a PDS.

Figure 6.2: Return Codes

Index